KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

UPC

eetac

Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

The Rician channel model for LAP-Ground communication

Sergi Marquez Andreu

Thesis submitted to
obtain the Bachelor's degree
in Telecommunication Systems

Supervisor:
Prof. Sofie Pollin

Mentors:
Mahdi Azari

Academic Year 2015 - 2016

Preface

I would like to thank to my supervisor, Prof. Sofie Pollin, for the opportunity to let me join in your research group and to live a rewarding experience in this university.

I would also like to express my gratitude to my mentor Mahdi for his support and patience especially initially of the semester period when I was going a bit lost.

Finally, the most important for me, my family: my father, my mother and my brother Albert, because without them I could not have come to Leuven to realize my bachelor thesis. Thank you very much to each of them for their support when I have needed some advice.

Sergi Marquez Andreu

Table of Contents

Abstract

In a communication channel between UAV-to-Ground in Low Altitude Platforms (LAP) there are many parameters to study like the fading. The fading effect occurs when the deviation or attenuation suffered a modulated signal through in certain telecommunications propagation media. Changes in position will cause both time dispersion of the signal and time variance of the channel, in this case appear which is known by small and large scale fading. Time dispersion of the signal leads to frequency selective or flat fading whereas time variance results in fast or slow fading. Several models are considered for small scale fading like the Rician fading, which occurs in *line-of-sight* conditions and Rayleigh fading derived in *non-line-of-sight*.

The Rician fading has an important parameter, the Rician K-factor, which determines the ratio of *line-of-sight* components to scattered waves. There are several methods to estimate the Rician K-factor: the moment's method, the maximum likelihood estimator and the fitting curve method.

The aim of this thesis is to see the behaviour of the Rician K-factor in a UAV-to-Ground communication link for different altitudes. The conclusions about which is the more accurate method and explain how the Rician K-factor varies for each altitude are explained.

List of figures

List of tables

List of abbreviations and symbols

UAV	Unmanned Aircraft Vehicles
HAP	High-Altitude Platforms
LAP	Low-Altitude Platforms
LMDS	Local Multipoint Distribution System
LoS	Line-of-sight
nLoS	non Line-of-sight
B_s	Signal bandwidth
B_c	Coherence bandwidth
B_d	Symbol bandwidth
Ds	Delay Spread
SNR	Signal-to-Noise ratio
ISI	Inter-Symbol Interference
T_c	Coherence time
T_s	Symbol duration
PDF	Probability Density Function
σ	Distribution mode
s	Field strength of the LoS component
I_0	Zeroth-order Bessel function
I_1	First-order Bessel function
K	Rician K-factor
MIMO	Multiple-input Multiple-output
HF	High Frequency
MLE	Maximum Likelihood Estimator
Γ	Gamma function
F_1	Confluent hypergeometric function
R(t)	Envelope
K_{in}	Input K-factor
$\widehat{K}_{out}$	Estimated K-factor
P_r	Received Power
V_r	Received Voltage
R_l	Load resistence
N	Number of samples

Chapter 1: Introduction

As we know, the first utility when we began to use the UAVs (Unmanned aircraft vehicles) was for military applications. The reason was because the idea that you can control the vehicles without any person was very interesting and USA was the first to experiment these applications in the First World War so they did not have human losses in battles.

The continuing evolutions of UAVs do not stop and they experiment many utilities in several areas for example in the quotidian live or in companies. The communications between from UAV to a ground can be robust for several reasons, either by signal fading or the path loss by distance for example.

It is true that in recent years the interest with the development of high-altitude platforms (HAP) has risen because since the idea to have a low cost stratospheric aircraft carrying payloads tailored for a wide range of applications in telecommunications and remote sensing is a great advance. These platforms are able to flying at altitudes between 17 and 30 Km with an autonomy that reaches several months.

One of the most promising applications of telecommunications HAPs are cellular telephony, local multipoint distribution system broadband services (LMDS), and providing access to digital networks (Internet, ISDN) [1] [2].

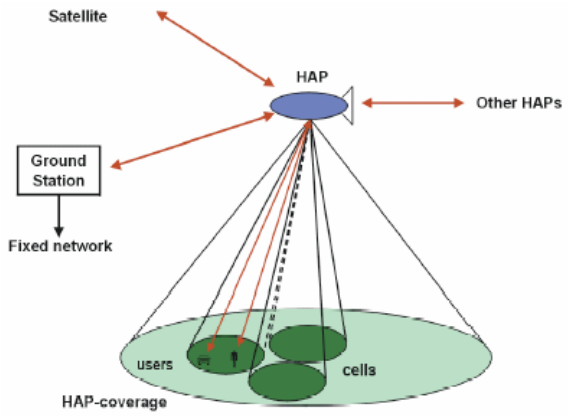


Figure 1.1: HAP network

The principal mobile network prospective for HAPs, such 3G systems, are available all over the world. Properly set, these networks have the capability of providing high performance wireless connectivity. [2]

On the other hand, appear another type which is known as the Low-Altitude Platforms (LAP). These platforms began to be used in Asia when there was some natural disaster like an earthquake or some tsunami. Where the traditional methods to manage these situations were ineffective, the communication systems using balloons over the ground were suggested to enhance a communication backbone. The Low-Altitude Platforms (LAP) was an alternative solution to emergency communication systems [3].

The entire environment between transmitter and receiver is the channel that it is an important thing. Vehicles and people in the city are moving permanently and all of this makes the channel are not stationary. There are two key important parameters for communications channel; one is the delay spread and the other, the type of fading. The delay spread holds information about the time dispersion the channel induces, all of this is reflected in the fading.

The exchange of data for common tactical large scale UAV-based systems is currently usually realized by means of propriety point-to-point connections. For links of LoS, if exist a *Line-of sight* between transmitter and receiver. The received signal consists of a significant line component direct view (main component) and multicomponent ray reflected (as in non-LoS configuration). In this case, the envelope of the received signal means a Rice distribution, and the fading suffering is classified as type distribution of Rician fading. On the other case, if we have a *non-line-of-sight* means Rayleigh fading. In the figure 1.2 we can see the difference between the LoS and nLoS. [4]

Figure 1.2: Example of LoS and nLoS channel

In a direct link, the power ratio between the LoS link and nLoS link can be increased if the width is reduced beam transmitter. This parameter is known as Rician K-factor The aim of this project is to study the behaviour of the Rician K-factor for different altitudes of the UAV. To do this, the document is structured as follows:

- Chapter 2 → Main concepts about the propagation channel and the study of the different methods to estimate Rician K-factor.
- Chapter 3 → Study of the empirical measurement carried out.
- Chapter 4 → Conclusions of the work done.

Chapter 2: Theory

In this chapter, we introduce several concepts necessary to know which elements are involved in a communication channel.

The chapter is divided into the following sections: 2.1 introduce the concept of fading. In 2.1.1 the frequency selective and flat fading is presented. In 2.1.2 explain the fast and slow fading concepts. In 2.2 present the probability density function and inside I will do a brief explanation of two types of distributions: 2.2.1 Rayleigh distribution, 2.2.2 Rician distribution. In 2.3 present the methods to estimate the Rician K-factor: 2.3.1 moment's method, 2.3.2 maximum likelihood method, 2.3.3 fitting curve method. Finally, in 2.4 we will do a comparison of the several methods.

2.1 Fading

In wireless communications, fading is the deviation or attenuation suffered a modulated signal through in certain telecommunications propagation media. The decrease can vary with time, geographical position and/or the radio frequency, and is modelled as a random process.

The presence of reflector in the environment of a transmitter and receiver to create multiple paths that can cross a transmitted signal. As a result, the receiver sees the superposition of the multiples copies of the transmitted signal, each should traverse a different path. Each copy of the signal will experiment differences in signal attenuation, delay and phase shift during travel from the source to the receiver. This can result in any of constructive or destructive interference, amplifying or attenuating the power of the sign seen from the receiver.

A common example of this, it is the experience of stopping in a semaphore and listening to FM's emission to degenerate in statically, whereas the sign is recovered if the vehicle moves itself only a fraction of a meter. The loss of the emission is caused because the vehicle stops in a point where the sign of destructive interference experiences faults. The cell phones also can present similar momentary fading.

Models of channel fading often are in use for shaping the effects of the electromagnetic transmission of the information about the air in the cellular

networks and of communication of diffusion. Models of channel fading also are in use in the acoustic submarine communication for shaping the distortion caused by the water. Mathematically, acts like a variable in the time, the change at random in the extent and the phase of the transmitted sign [5].

Changes in position will cause both time dispersion of the signal and time variance of the channel. There are two types:

- Small-scale fading, this happens in order of the carrier wavelength and frequency dependent. The small-scale fading causes the most severe problems.
- Large-scale fading, this happens as the mobile moves through a distance of the order of the cell size, and is typically frequency independent.

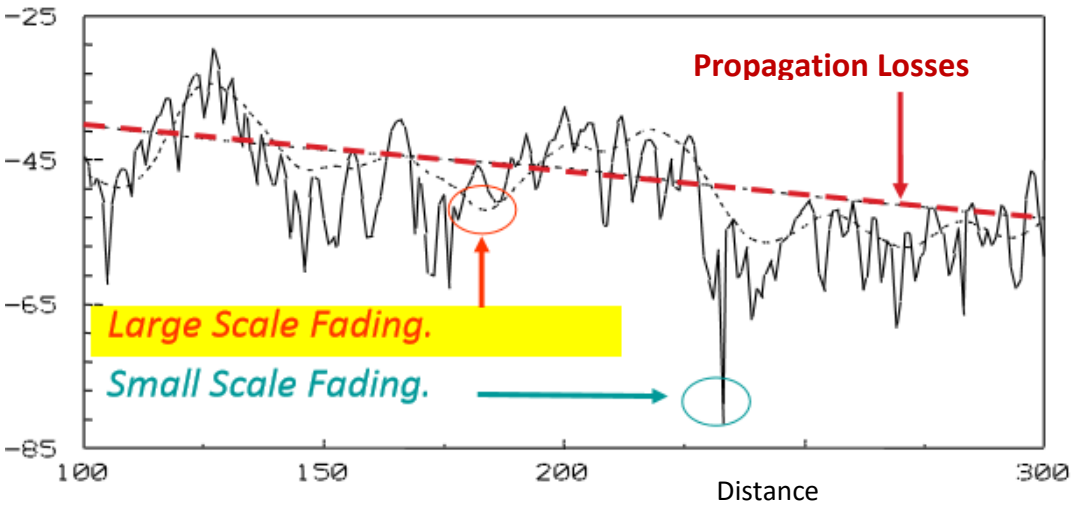


Figure ¡Error! No hay texto con el estilo especificado en el documento.**2.1 Small and scale fading**

Time dispersion of the signal leads to frequency selective or flat fading whereas time variance results in fast or slow fading. These concepts are explained in the next sections.

2.1.1 Frequency selective and flat fading

The wireless channel is said to be flat fading if it has constant gain and linear phase response over a bandwidth which is greater than the bandwidth of the transmitted signal. In other words, flat fading occurs when the bandwidth of the transmitted signal B_s is smaller than the coherence bandwidth of the channel B_c, when we speak about the coherence bandwidth in the frequency domain, is used to denote the range of the frequencies over which a channel affects passing signals with equal gain and

linear phase. This bandwidth is approximately the inverse of the maximum excess delay $\rightarrow B_c < \frac{1}{T_s}$

$$B_s \ll B_c \equiv \frac{1}{Ds} \gg \frac{1}{T_{simb}} \equiv Ds \gg T_{simb} \quad (1)$$

The effect of flat fading channel can be seen as a decrease of the Signal-to-Noise Ratio (SNR). Since the signal is narrow with respect to the channel bandwidth, the flat fading channels are also known as amplitude varying channels or narrowband channels. [5][6][7]

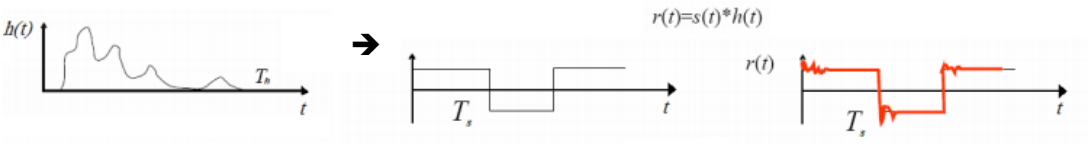


Figure 2.2: Flat fading channel

But if we have the other condition, and the bandwidth of the transmitted signal is bigger than the coherence bandwidth the channel is distorted, not preserve the original spectrum of the transmitted signal and is called frequency selective channel. The conditions are the following:

$$B_s \gg B_c \equiv \frac{1}{DS} \gg \frac{1}{T_{simb}} \equiv Ds \gg T_{simb} \quad (2)$$

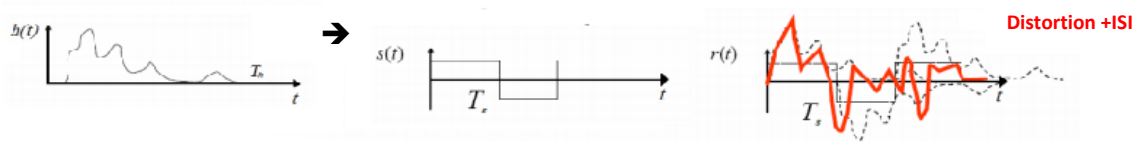


Figure 2.3: Frequency selective channel

2.1.2 Fast and slow fading

Fast fading or slow fading relate to the variations in channel response with respect to time. When channel response changes at very short intervals, then it is called fast fading. On the other hand, when the channel response changes at long intervals, then it is called slow fading. Speed of an object determines fast or slow fading. Speed indicates the rate of change in the channel responds with respect to time. The coherence time T_c of a channel is defined as the duration for which the channel remains unchanged. High speed indicates high rate of change and therefore the duration for which channel remains static is small, indicating small coherence time T_c. All of this causes fast fading, which takes places when the coherence time of the channel is smaller compared to the symbol duration T_s . In fast fading, the channel impulse response may change several times within the symbol period. In frequency domain high speed results in high amount of frequency spreading, which will be more than symbol bandwidth [8]. The fast fading occurs in the conditions when:

$$T_c \ll T_s \equiv B_d \gg B_s \tag{3}$$

Slow speed indicates slow rate of changes and therefore the duration for which channel remains static is long, indicating large coherence time T_c. This causes slow fading, which takes place when the coherence time of the channel is larger compared to the symbol duration. In slow fading, the channel impulse response may remain same over several symbol periods. In frequency domain, slow speed results in low amount of frequency spreading, which will be less than symbol bandwidth [8]. Combining the frequency domain as well as time domain consequences, the slow fading occurs in the conditions when:

$$T_c \gg T_s \equiv B_d \ll B_s \tag{4}$$

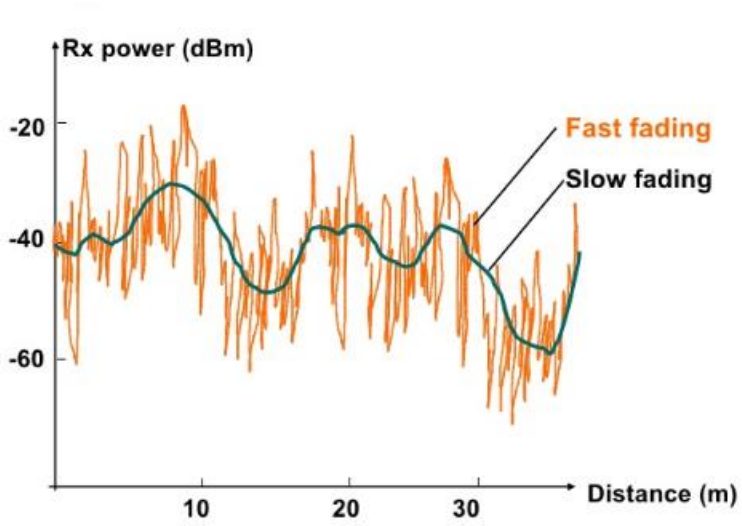


Figure 2.4: Fast and slow fading

2.2 Probability density function

The channel can be effected by time dispersion, multipath propagation or type of fading.. In the dense environments like the urban environments wher you can find many buildings can produce many scattered waves. Therefore, it is difficult to obtain a *line-of-sight,* LoS, communication between transmitter i receiver in this type of environments. On the other hand, there are areas with less density of building like a rural environment that introduces less scattering and it is possible to obtain *line-of-sight* communication.

The receive signal is composed of a mixture of delayed, reflected, and scattered waves, this effects can be reflect in the amount of each type of received waves. These variations will affect the statistics of the fading

We have a lot of statics propagations models in communication systems like Nakagami-m fading model, Weibull fading model or Lognormal fading model, but I only will focus in two types: Rayleigh fading model and Rician fading model.

2.2.1 Rayleigh fading model

When there are a large number of reflections in an environment, the name given is Rayleigh fading.. The type of fading model uses a statistical approach to analyze the propagation, and can be used in a number of environments.

The Rayleigh fading model is normally viewed in a densely urban environment where there are many reflections and scattered waves from buildings, trees, cars, etc.

The Rayleigh fading model can be used to analyze radio signal propagation on a statistical basis. It operates best under conditions when there is no dominant signal when we speak about this term we say *non-line-of-sight*, nLoS*,* and in many instances cellular telephones being used in a dense urban environment fall into this category [9].

The Rayleigh probability density function is defined for the following expression:

$$f(r) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \qquad for\ r \geq 0 \tag{5}$$

where σ is the distribution mode and r is a random variable, which actually represents the magnitude of the complex envelope [10].

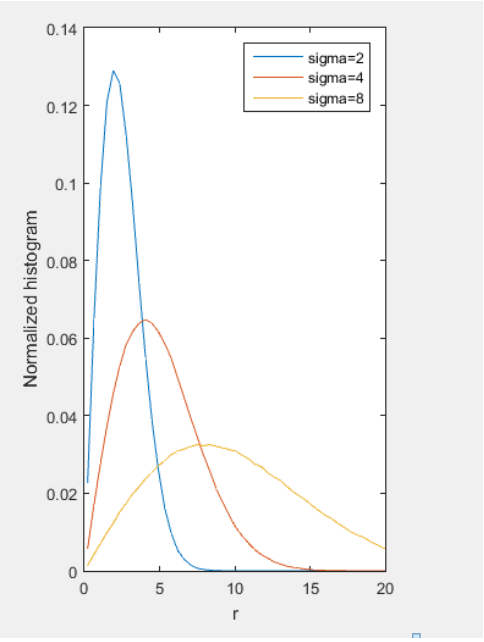


Figure 2.5: Rayleigh distribution with several sigma

2.2.2 Rician fading model

Rician fading model is the most important in my thesis because I will study his behavior. The model behind Rician fading is similar to the Rayleigh fading except that in Rician fading a strong dominant component is present. This dominant component can for instance be the *line-of-sight* wave. Since this wave often is strong compared to the scattered waves, the *PDF* of the amplitude will change.

This new form is the Rician distribution defined as

$$f(r) = \frac{r}{\sigma^2} exp^{-\frac{(r^2+s^2)}{2\sigma^2}} I_0\left(\frac{rs}{\sigma^2}\right) \qquad for\ r \geq 0 \qquad (6)$$

where s is the field strength of the LoS component and I_0 is the zero-order modified Bessel function of the first kind.

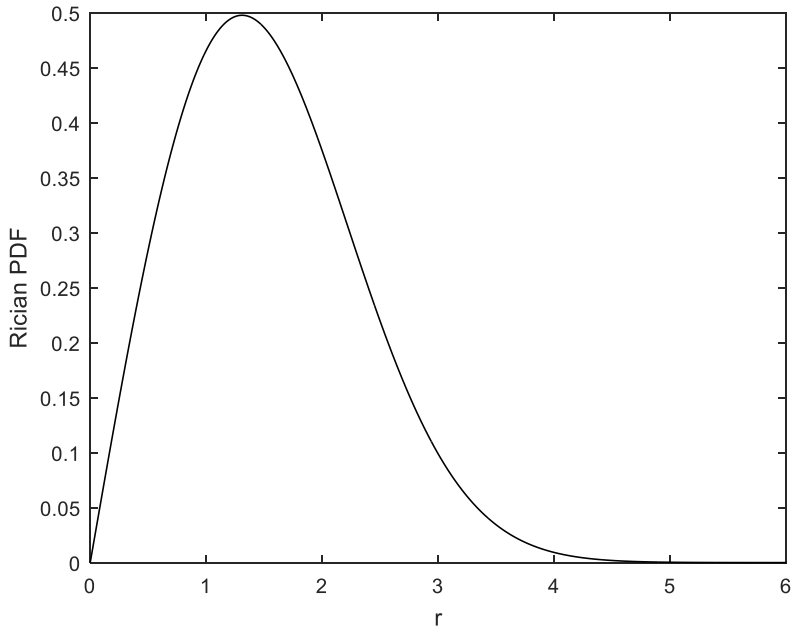


Figure 2.6: Rician distribution with s=1 and σ=1

In the Rician channel there is an important parameter which is the Rician K-factor where is defined as

$$K = \frac{s^2}{2\sigma^2} \tag{7}$$

expresses the ratio dominant component to the scattered waves. In fact, as K→ ∞, the Rician PDF → $Gaussian$ and as $K \to 0$, the Rician PDF → Rayleigh. That is, the stronger the *line-of-sight* component is, the greater will the shift of mean be for the scattered waves. Such a shift will make the Rician distribution approach Gaussian distribution. As the direct wave weakens the shift of mean will approach zero and the Rician PDF becomes Rayleigh [4].

We can write the PDF in terms of K-factor (dB)

$$f(r) = \frac{2r10^{K/10}}{s^2} e^{-\frac{10^{K/10}}{s^2}(r^2+s^2)} I_0\left(\frac{2r10^{K/10}}{s^2}\right) \quad for\ r \geq 0 \tag{8}$$

and we can see in the following graphs the behavior explained:

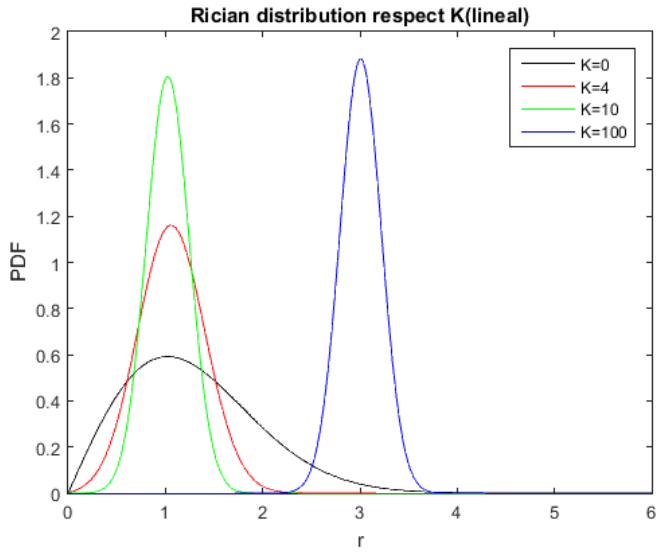


Figure 2.7: Rician distribution in terms of K-factor

2.3 Parameter estimation

One form to measure the quality in a communication channel is through the K-factor, which show the relative power of the LoS component. Therefore, estimation of K is of practical importance in a several situations of wireless scenarios like link budget calculations or geolocation applications. Besides, many advances have found that the Rician K-factor has a great influence on the capacity and performance of multiple-input multiple-output (MIMO) in the coding. Hence, estimation of the Rician factor is important not only for channel characterization, but also in adaptive modulation schemes, where accurate estimates and the knowledge of the estimation error is crucial for proper system operation [11].

There are some methods to carry out to estimate the Rician K-factor. For example, Rastogi and Holt propose a moment based approach which utilizes the second- and fourth-order moments of the envelope in order to estimate from the high frequency (HF) radio waves. Another method is the maximum likelihood estimator (MLE) is derived. A simpler estimator that utilizes the first and second order moments of the received envelope, which also requires the inversion of a nonlinear function of K, is proposed in many papers. The distribution fitting approaches for estimating K, are robust, but are not suited for online implementation due to their complexity. An expectation-maximization approach to finding the MLE for a multidimensional Rician distribution but still not easy to calculate and use in a communication scenario [11].

The received data is presumed to be independent and identically distributed in several of these methods. In this section we focus in four types to estimate K: first and second order moments, second and fourth order moments, maximum likelihood estimator and fitting curve method. For these methods, we are going to design algorithms in Matlab program to estimate the Rician K-factor to verify the correctly method.

2.3.1 Method of moments

The advantage to use the method of moment's estimator is easy to find and to implement. This method is more accurate as long as the data set is large.

This method is based on the moments of a PDF. The moments of the Rician distribution, expressed in terms of σ^2 and K, are given by

$$\mu_n \coloneqq E[R^n(t)] = (\sigma^2)^{n/2}\Gamma\left(\frac{n}{2}+1\right)\exp(-K)\,\mathrm{F}_1\left(\frac{n}{2}+1;1;K\right) \tag{9}$$

where $\mathrm{F}_1(\cdot\,;\cdot\,;\cdot)$ is the confluent hypergeometric function, and $\Gamma(\cdot)$ is the gamma function. We can see from (9) that the moments depend on the two unknown parameters σ and K. Hence, a moment-based K estimator requires estimates of a least two different moments of R(t). More specifically, suppose that for $n \neq m$ we define the following functions of K

$$f_{n,m}(K) \coloneqq \frac{\mu_n^m}{\mu_m^n} \tag{10}$$

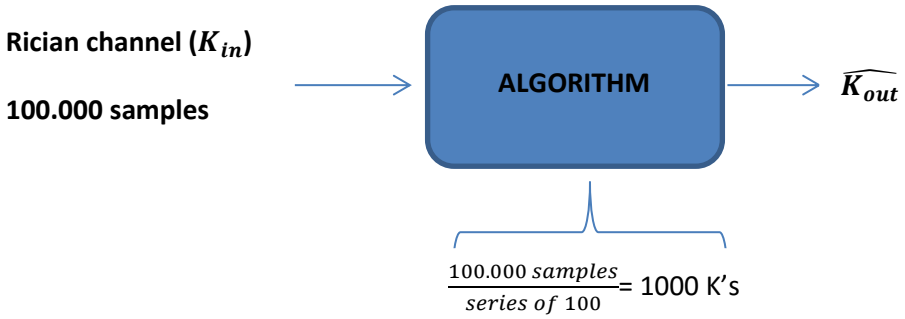
where μ_n is the nth moment of R(t). Since by the (9) and (10) $f_{n,m}(K)$ depends only on K and not on σ, we can construct moment-based estimator for K by using sample moments instead of the ensemble values in (10) and then inverting the corresponding $f_{n,m}(K)$ to solve for K. Hence, an estimator that depends on the mth and nth moments could be expressed as

$$\widehat{K}_{n,m} \coloneqq f_{n,m}^{-1}\left(\frac{\hat{\mu}_n^m}{\hat{\mu}_m^n}\right), \qquad with\ \mu_k \coloneqq \frac{1}{N}\sum_{l=0}^{N-1} R^k(lT_s) \tag{11}$$

where N is the number of available sample and T_s is the sampling period and we assume that the inverse function $f_{n,m}^{-1}(\cdot)$ exists. For all the values of m and n we considered $f_{n,m}(K)$ is a monotone increasing function in the interval$K \in (0,\infty)$, and hence, the inverse function $f_{n,m}^{-1}(\cdot)$ does exist.
The natural choice for (n,m) is (1,2) since this selection involves the lowest order moments [11]. When n=1 and m=2, K can be calculated using the following equation

$$f_{1,2}(K) = \frac{\pi e^{-K}}{4(K+1)}\left[(K+1)I_0\left(\frac{K}{2}\right)+KI_1\left(\frac{K}{2}\right)\right]^2 \tag{12}$$

All of these equations are used to calculate the first and second moment based. To verify this method, as we said, we have simulated in Matlab program the algorithm of first and second moment. First of all, several Rician channels are generated (with a hundred thousand of samples) between a transmitter and receiver with different values of K's, then all the samples of the envelope are introduced into the algorithm designed, inside of the algorithm the samples are separated in packages of series of 100, so finally several values of K's are obtained and their respective PDF's are showed. We have to demonstrate that the input K is similar to the output K. In the next figure an overview is presented:



The algorithm has been designed with values of $K_{in} = 0, K_{in} = 4,\ K_{in} = 20$ and $K_{in} = 100$. In the next figures we see the PDF generated by the algorithm for the different values of K_{in}.

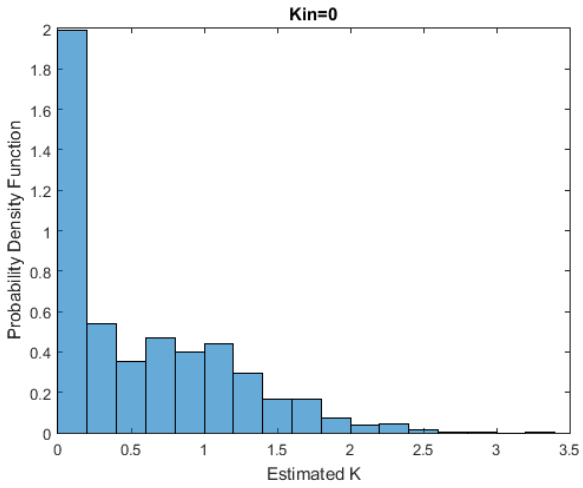


Figure 2.8: 1st and 2nd moment: Rician distribution with $K_{in} = 0$

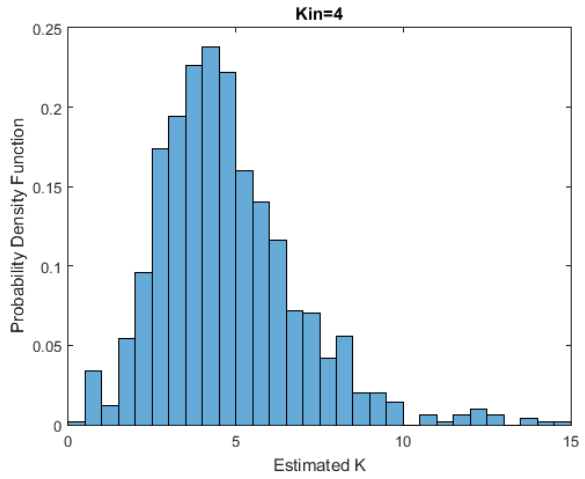


Figure 2.9: 1st and 2nd moment: Rician distribution with $K_{in} = 4$

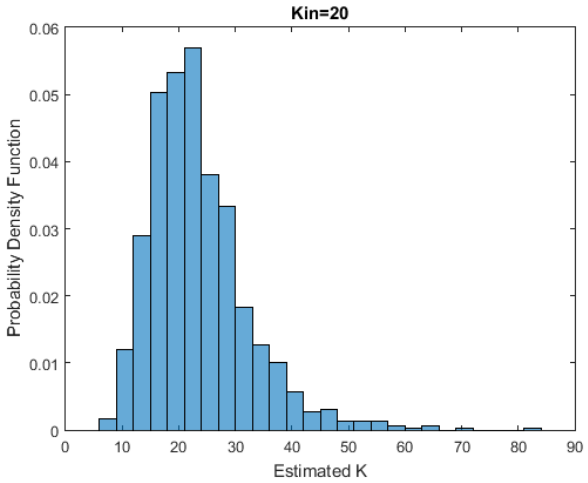


Figure 2.10: 1st and 2nd moment: Rician distribution with $K_{in} = 20$

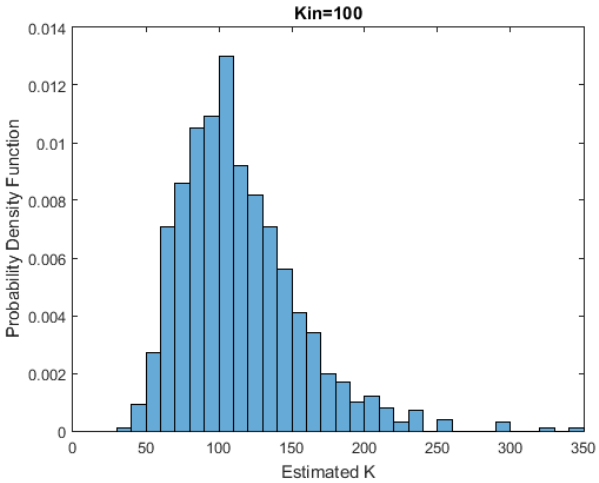


Figure 2.11: 1st and 2nd moment: Rician distribution with $K_{in} = 100$

A simple alternative to the first and second moment is to calculate the second and fourth moment [11] [12]. It can be shown using the (9) and (10) that

$$f_{(2,4)}(K) = \left[\frac{(K+1)^2}{K^2+4K+2}\right]^2 \tag{13}$$

Solving for $\widehat{K}_{2,4}$ from an estimate of the left-hand side in (12) involves finding the roots of a second-order polynomial which can be done in closed form. It can be shown that one of the roots of this polynomial is always negative which can be discarded since $K \geq 0$, yielding a unique non negative solution for $\widehat{K}_{2,4}$ which is given by

$$\widehat{K}_{2,4} = \frac{-2\hat{\mu}_2^2 + \hat{\mu}_4 - \hat{\mu}_2\sqrt{2\hat{\mu}_2^2 - \hat{\mu}_4}}{\hat{\mu}_2^2 - \hat{\mu}_4} \tag{14}$$

We can calculate $\hat{\mu}_2$ and $\hat{\mu}_4$ from the equation (11).

For this method the procedure is repeated like in the first and second moments to calculate the PDF to estimate $\widehat{K}_{2,4}$

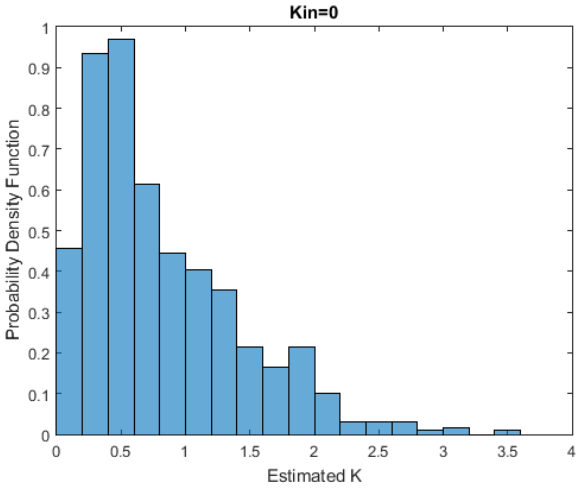


Figure 2.12: 2nd and 4th moment: Rician distribution with $K_{in} = 0$

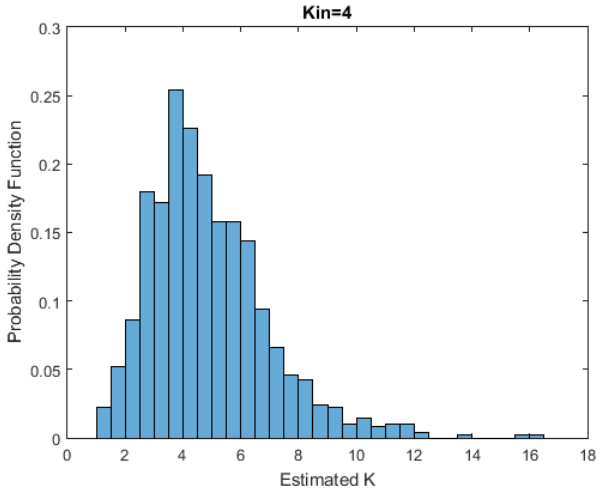


Figure 2.13: 2nd and 4th moment: Rician distribution with $K_{in} = 4$

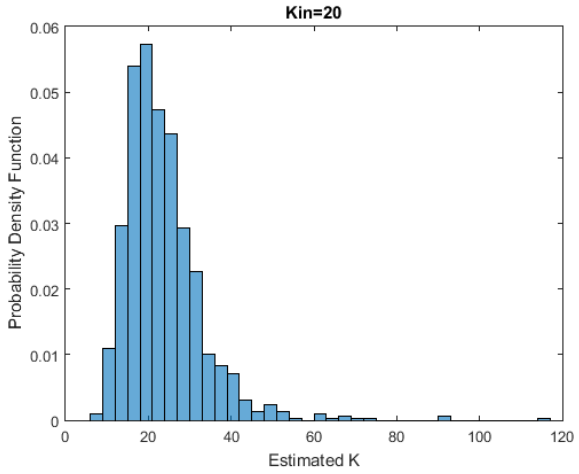


Figure 2.14: 2nd and 4th moment: Rician distribution with $K_{in} = 20$

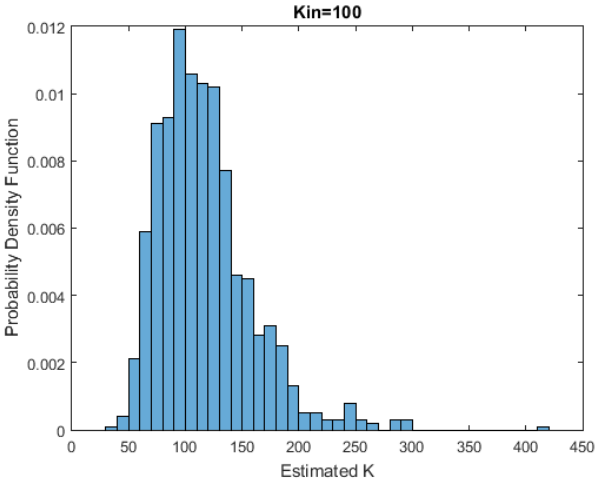


Figure 2.15: 2nd and 4th moment: Rician distribution with $K_{in} = 100$

2.3.2 Maximum likelihood estimation (MLE)

This method is one the most popular as it can be implemented to estimate different parameters and for large data sets the variance of the estimator error is reduced and can improve the estimation.

The MLE exploits the a priori knowledge of the data statistics in an optimal way. Moreover, when the number of data samples increases, the distribution of the MLE approximate to the normal distribution.

The joint PDF of a sample of N independent observations $\{r_i\}$ is called the likelihood function of the sample, and is written as

$$L = \prod_{i=1}^{N} p(r|s) \tag{15}$$

where $p(r|s)$ is given in the rician distribution. The MLE can be constructed directly from the likelihood function L. Finding the parameters that are most likely for the given set of samples is done by finding the maximum of this likelihood function. To make the calculations easier we take the natural logarithm of L and get:

$$logL = log \prod_{i=1}^{N} \frac{2r10^{K/10}}{s^2} e^{-\frac{10^{K/10}}{s^2}(r^2+s^2)} I_0\left(\frac{2r10^{K/10}}{s^2}\right) \tag{16}$$

In order to derive the MLE estimator of Rician K-factor, the logarithm is derived with respect K. Setting the derivate to, the estimated Rician K-factor can be obtained by

$$\widehat{K} = \frac{N}{2} \frac{|s|^2}{\sum_{i=0}^{N-1}(|s_i| - |s|)^2} \tag{17}$$

where $|s|^2$ is constant and is considered to be the mean of direct component of the data samples. While $|s_i|$ is i-th the electrical field data sample [13] [14].

With the same procedure of the others methods the estimated Ks are calculated with this method to different values of K_{in}.

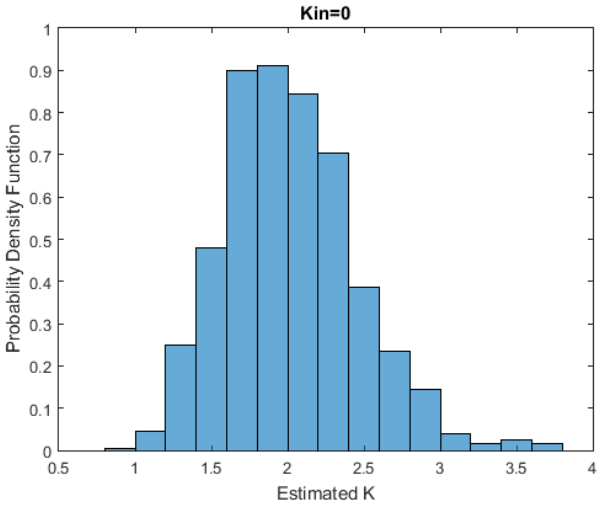


Figure 2.16: MLE: Rician distribution with $K_{in} = 0$

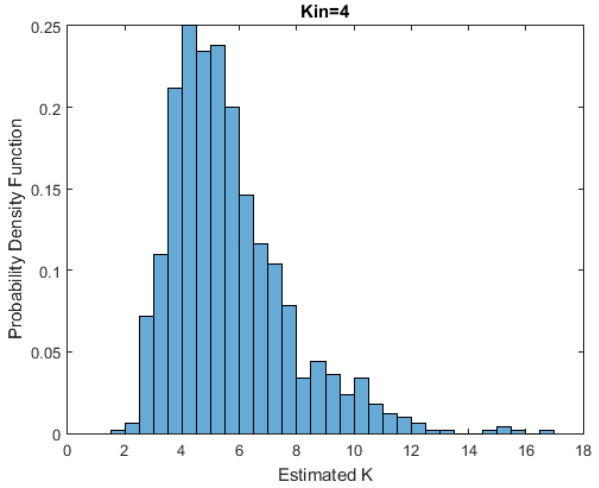


Figure 2.17: MLE: Rician distribution with $K_{in} = 4$

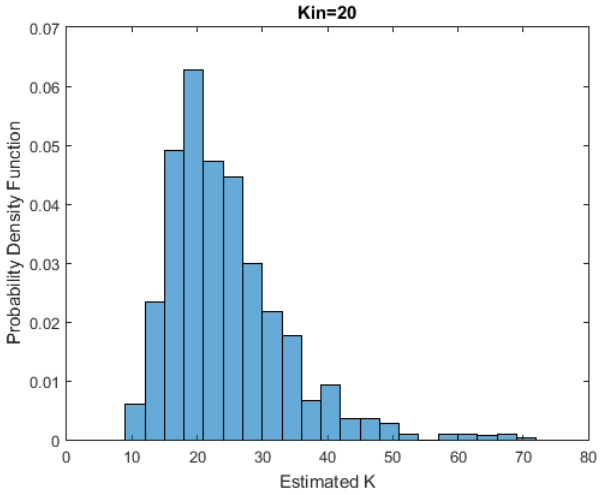


Figure 2.18: MLE: Rician distribution with $K_{in} = 20$

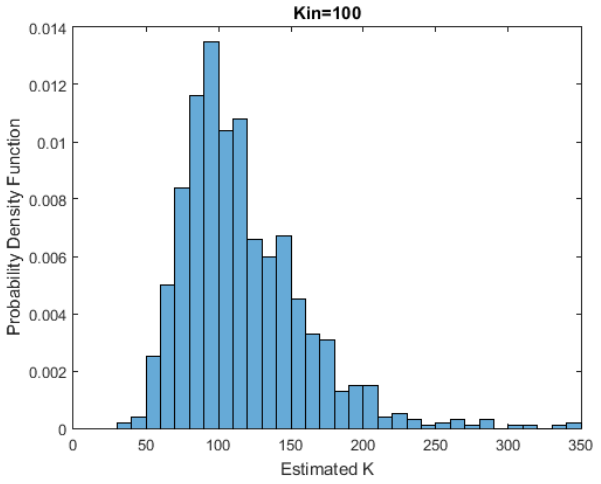


Figure 2.19: MLE: Rician distribution with $K_{in} = 100$

2.3.3 Fitting curve method

The aim of this method is to obtain the parameters of a model that describes a set of data in a mode to reduce the difference between the model and the data. Computes the probability distribution of the measured data and compares the result to a set of hypothesis distributions using a suitable goodness-of-fit test.

Then the Rician K-factor is given by

$$\widehat{K} = \frac{s_I^2 + s_Q^2}{2\sigma^2} \tag{18}$$

where s_I^2 and s_Q^2 is the received voltage with I/Q components. Hence, calculate the estimate K for several series: first is calculating the mean of the samples in a series and then, divided for the variance of this series [15] [16] [17]. With the application distribution fitting of Matlab programs the values of estimated K's are the following:

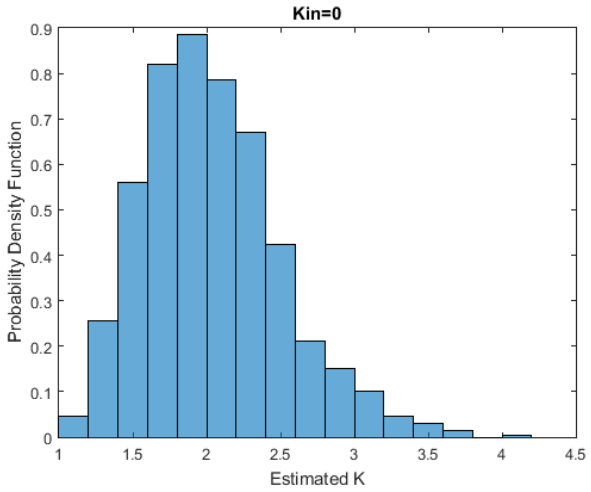


Figure 2.20: Fitting curve: Rician distribution with $K_{in} = 0$

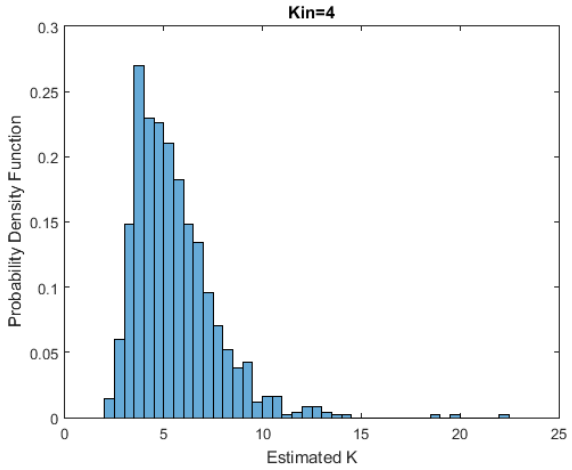


Figure 2.21: Fitting curve: Rician distribution with $K_{in} = 4$

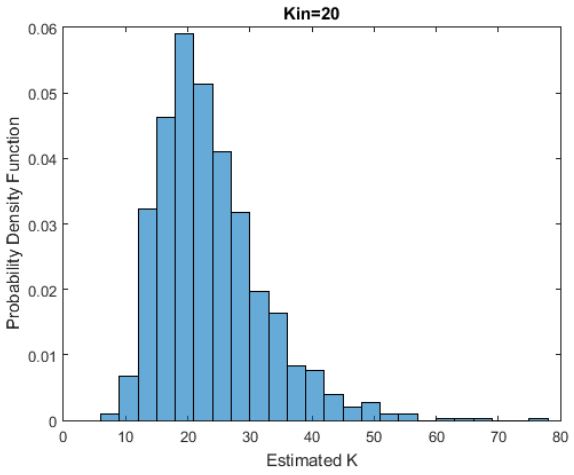


Figure 2.22: Fitting curve: Rician distribution with $K_{in} = 20$

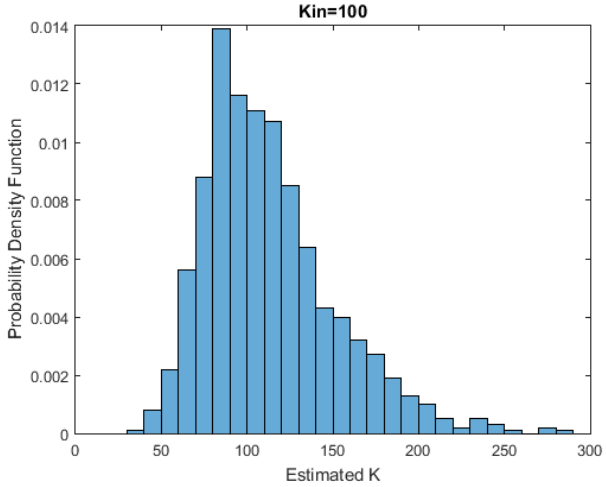


Figure 2.23: Fitting curve: Rician distribution with $K_{in} = 100$

2.4 Comparison of methods

Two moment-based estimators $\widehat{K}_{1,2}$ and $\widehat{K}_{2,4}$ were worthy of special attention because $\widehat{K}_{1,2}$ had the best asymptotic performance, and $\widehat{K}_{2,4}$ had a simple closed-form expression in terms of the moments. The simulations corroborate that the moment-based estimators that use the envelope are very close to MLE even for finite sample size. On the following tables we will see the difference between the values that we obtained and we can ask the question about what is the best estimator.

	1st and 2nd moment	2nd and 4th moment	MLE	Fitting curve
$\mathbf{K_{in} = 0}$	$\widehat{K}_{out} = 0.56$	$\widehat{K}_{out} = 0.82$	$\widehat{K}_{out} = 2.01$	$\widehat{K}_{out} = 2.02$
$\mathbf{K_{in} = 4}$	$\widehat{K}_{out} = 4.74$	$\widehat{K}_{out} = 4.88$	$\widehat{K}_{out} = 5.6$	$\widehat{K}_{out} = 5.45$
$\mathbf{K_{in} = 20}$	$\widehat{K}_{out} = 23.5$	$\widehat{K}_{out} = 23.64$	$\widehat{K}_{out} = 23.87$	$\widehat{K}_{out} = 23.95$
$\mathbf{K_{in} = 100}$	$\widehat{K}_{out} = 116.46$	$\widehat{K}_{out} = 117.17$	$\widehat{K}_{out} = 115.53$	$\widehat{K}_{out} = 115.71$

Table 2.1: Estimate K's in the different methods

We can see that for small K's the more accurate method are the moment's methods. On the other hand, for large values of K's we can see that the better method is maximum likelihood method followed by fitting curve method but all the methods are closed.

All the methods are biased estimators for finite data samples and for calculate the bias samples, the following equation is used [11]:

$$bias = \frac{1}{N}\sum_{N=1}^{N} \widehat{K(n)} - K \tag{19}$$

The following table, the respective bias are calculated

	1st and 2nd moment	2nd and 4th moment	MLE	Fitting curve
$\mathbf{K_{in} = 0}$	bias=0.56	bias=0.82	bias=2.01	bias=2.02
$\mathbf{K_{in} = 4}$	bias=0.74	bias=0.88	bias=1.6	bias=1.45
$\mathbf{K_{in} = 20}$	bias=3.5	bias=3.64	bias=3.87	bias=3.95
$\mathbf{K_{in} = 100}$	bias=16.46	bias=17.17	bias=15.53	bias=15.71

Table 2.2: Bias samples in each method

We can take a conclusion about the table 2.2 and it is that when the Rician K-factor increase, the bias also [11].
For the next tables we will see the behavior of the methods when the number of samples and series are changed. After all, the conclusions about that are taken.

1st and 2nd moment

	N=1000 and series of 20	N=1000 and series of 50
$K_{in} = 0$	$K_{out} = 1.56$	$K_{out} = 0.76$
$K_{in} = 4$	$K_{out} = 13.44$	$K_{out} = 4.47$
$K_{in} = 20$	$K_{out} = 38.31$	$K_{out} = 34.77$
$K_{in} = 100$	$K_{out} = 162.03$	$K_{out} = 15855$

N=10.000 and series of 20	N=10.000 and series of 50
$K_{out} = 1.53$	$K_{out} = 0.73$
$K_{out} = 9.76$	$K_{out} = 5.71$
$K_{out} = 56.32$	$K_{out} = 28.09$
$K_{out} = 1.56$	$K_{out} = 134.62$

Table 2.3: Comparison for small and large N in 1st and 2nd moment method

2nd and 4th moment

N=1000 and series of 20	N=1000 and series of 50
$K_{out} = 2.4$	$K_{out} = 1.32$
$K_{out} = 12.24$	$K_{out} = 5.99$
$K_{out} = 46.73$	$K_{out} = 31.39$
$K_{out} = 204.87$	$K_{out} = 112.77$

N=10.000 and series of 20	N=10.000 and series of 50
$\boldsymbol{K_{out} = 2.2}$	$K_{out} = 1.17$
$\boldsymbol{K_{out} = 10.63}$	$K_{out} = 5.4$
$\boldsymbol{K_{out} = 41.71}$	$K_{out} = 28.72$
$\boldsymbol{K_{out} = 235.52}$	$K_{out} = 134.23$

Table 2.4: Comparison for small and large N in 2nd and 4th moment method

Maximum Likelihood Estimator

N=1000 and series of 20	N=1000 and series of 50
$\boldsymbol{K_{out} = 2.66}$	$K_{out} = 2.19$
$\boldsymbol{K_{out} = 9.57}$	$K_{out} = 6.6$
$\boldsymbol{K_{out} = 38.4}$	$K_{out} = 33$
$\boldsymbol{K_{out} = 313.53}$	$K_{out} = 132.59$

N=10.000 and series of 20	N=10.000 and series of 50
$\boldsymbol{K_{out} = 3.07}$	$K_{out} = 2.24$
$\boldsymbol{K_{out} = 10.89}$	$K_{out} = 6.11$
$\boldsymbol{K_{out} = 55.71}$	$K_{out} = 26.62$
$\boldsymbol{K_{out} = 257.23}$	$K_{out} = 137.43$

Table 2.5: Comparison for small and large N in MLE

Fitting curve

N=1000 and series of 20	N=1000 and series of 50
$\boldsymbol{K_{out} = 2.6}$	$K_{out} = 2.04$
$\boldsymbol{K_{out} = 8.48}$	$K_{out} = 6.84$
$\boldsymbol{K_{out} = 49.13}$	$K_{out} = 27.29$
$\boldsymbol{K_{out} = 227.8}$	$K_{out} = 151.26$

N=10.000 and series of 20	N=10.000 and series of 50
$\boldsymbol{K_{out} = 2.71}$	$K_{out} = 2.12$
$\boldsymbol{K_{out} = 10.71}$	$K_{out} = 6.47$
$\boldsymbol{K_{out} = 47.09}$	$K_{out} = 27.9$
$\boldsymbol{K_{out} = 218.31}$	$K_{out} = 130.31$

Table 2.6: Comparison for small and large N in fitting curve method

For small values of K's, we can use series of 20 but for large values of K's it is better if we use a series of ≥ 50 because if not, the error is significant. It is true that if we put a series over 50 all the estimated K's improve in all the methods.

In addition, the behavior in the methods is close but there are some methods more accurate and the error is smaller. This can be seen in the table 2.1.

Chapter 3: Empirical measurements

The aim of this chapter is to see the behavior of the UAV and balloon at different altitudes and angles to calculate the path loss and Rician K-factor. This chapter is divided in different scenarios: the first scenario a UAV has been used and for the second scenario a balloon.
To estimate the Rician K-factor the moment's methods have been used in all the scenarios.

3.1 First Measurements

In the first measurements chapter the first scenario is presented with the components and their respective simulations.

3.1.1 Requirements for real measurements in the first scenario

In this subchapter, we present the components and his respective characteristics to realize the real measurements with the UAV.
The transmitter module used was the RTC6705, from RichWave. This chip includes several bands. For the measurements done, a carrier frequency of 869 MHz was selected.

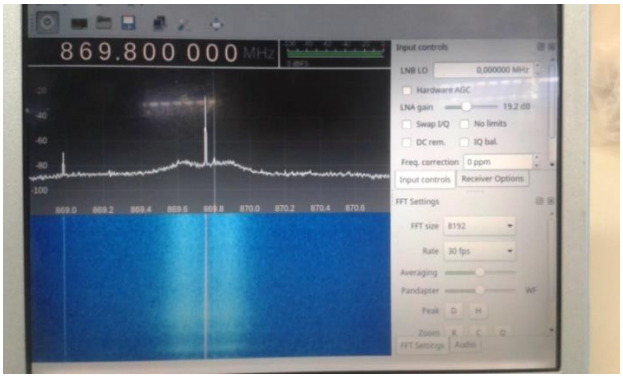


Figure 3.1: 869 MHz Rx Spectrums

The chosen UAV was the model IRIS+ from 3DRR, whose weight, payload included, was 0.890 Kg. It could be controlled either by an ISM 2.4 GHz link (i.e. manual mode) or by a predefined set of waypoints loaded through software (i.e. automatic mode).

Figure 3.2: UAV and controller

Finally, the receiver unit used was the USRP-2953R, from Ettus Research and the software to process the collected data was the GNU Radio.

3.1.2 Scenario and simulations with the UAV

In the first scenario the measurements were taken in the roof of the ESAT building. The UAV was flying to different altitudes that were 0, 4, 6, 8 and 10 meters and in each altitude, the measurements were taken. The UAV always had *line-of-sight* with the ground station. In the following figure it is possible to see how it was the scenario:

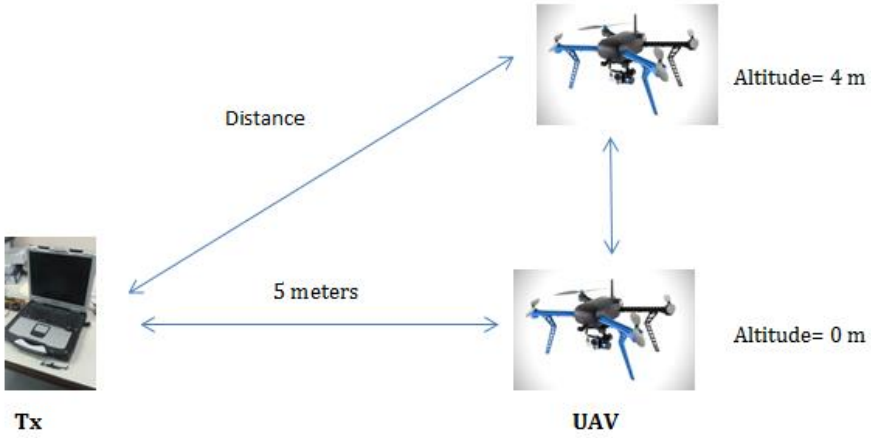


Figure 3.3: First scenario in the roof of the ESAT building

To calculate each real distance between the transmitter and receiver uses the Pythagoras theorem:

$$Distance = \sqrt{x^2 + Altitude^2} \tag{19}$$

where x is the distance between the transmitter and receiver in the ground and always will be 5 meters.

The first thing is to calculate the power received signal. The measurements are in microvolts because we have received the envelope signal and we need the power. The received voltage V_r has to be converted into power P_r in Watts through the equation below. A load resistance R_l of 50 Ω has been used.

$$P_r = \frac{V_r^2}{R_l} \tag{20}$$

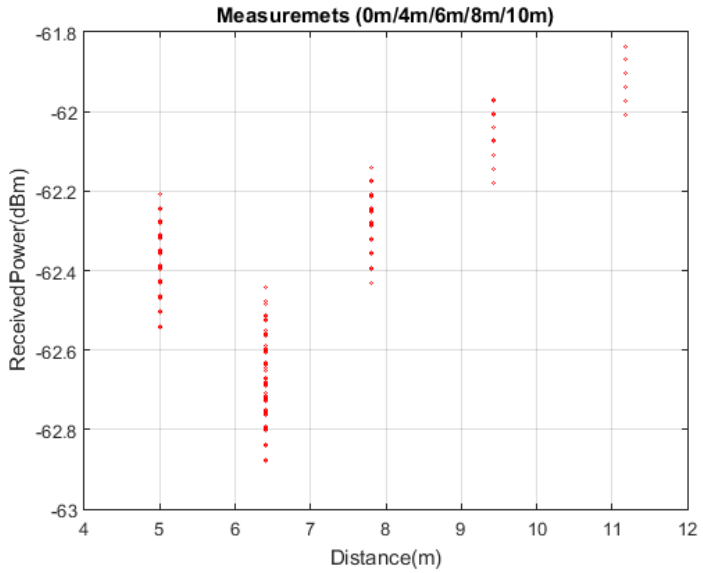


Figure 3.4: Received power in each altitude

It can be seen that for a large altitudes the received power not decrease. The reason about that, it might be to cause of the constructive or destructive fading. Therefore, it is possible in the altitude of 6 meters that the multipath effect causes constructive fading and the received power is bigger than in the altitude of 4 meters. The same reason occurs in the altitudes of 8 and 10 meters. It is true that the altitudes were close and the variation of the received power is minimal.

To calculate the estimate K of the received voltage, will be used the algorithms explained in the chapter of theory: 1st and 2nd moments, 2nd and 4th moments because there are the more accurate methods.

1st and 2nd moment

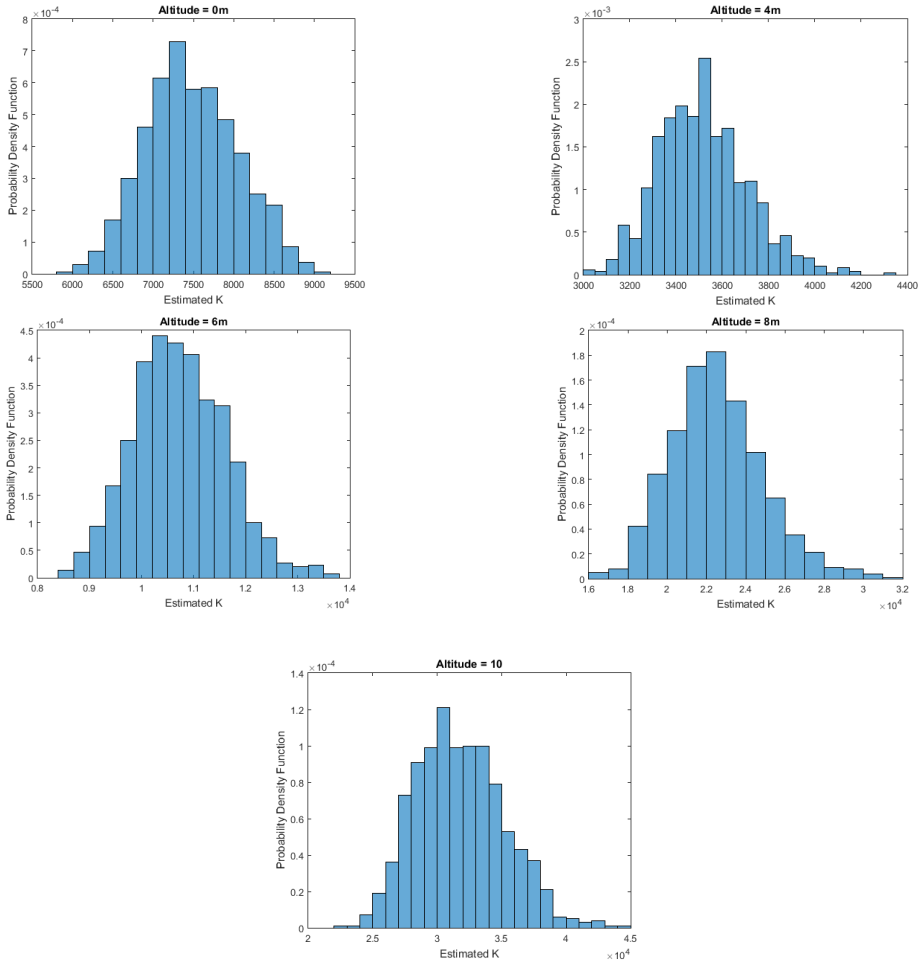
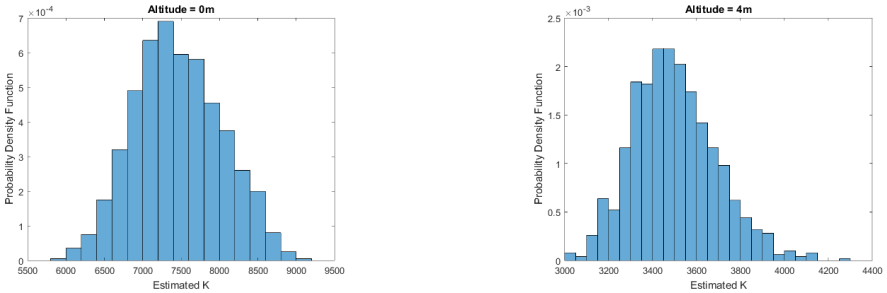


Figure 3.5: Estimated K's in 1st and 2nd moments for each altitude

2nd and 4th moment



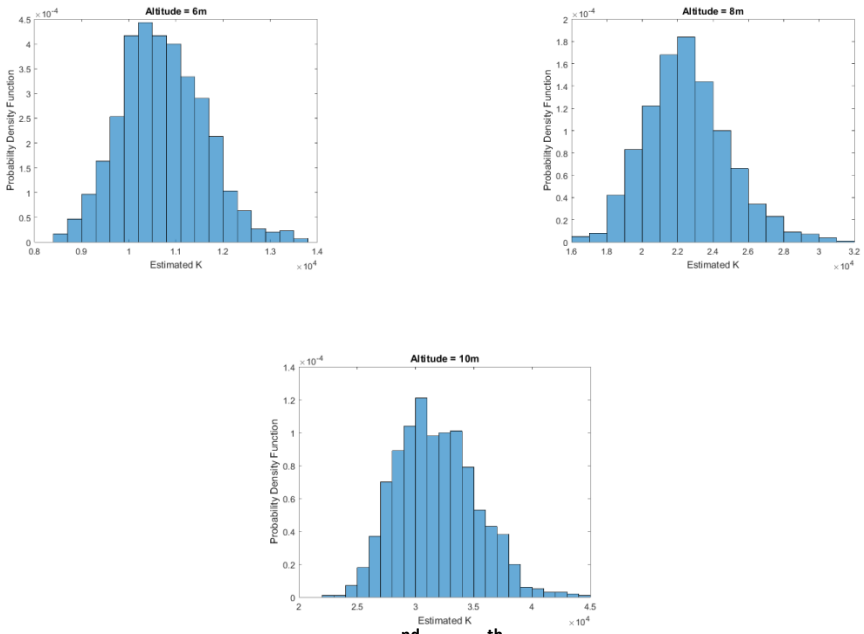


Figure 3.6: Estimated K's in 2nd and 4th moments for each altitude

Altitude (meters)		1st & 2nd moments	2nd & 4th moments
0	$\widehat{K}_{out}$	$7.4811 \cdot 10^3$	$7.4624 \cdot 10^3$
4	$\widehat{K}_{out}$	$3.5177 \cdot 10^3$	$3.4993 \cdot 10^3$
6	$\widehat{K}_{out}$	$1.076 \cdot 10^4$	$1.0712 \cdot 10^4$
8	$\widehat{K}_{out}$	$2.2548 \cdot 10^4$	$2.2548 \cdot 10^4$
10	$\widehat{K}_{out}$	$3.1784 \cdot 10^4$	$3.1795 \cdot 10^4$

Table 3.1: Estimate K's for each method in the first scenario

In the table of the estimated K we can see that the values are large. Because for lower altitudes the distance between the transmitter and receiver is short and consequently the direct power is bigger.

3.2 Second Measurements

In the second measurements chapter the different scenarios are presented with the components and their respective simulations.

3.2.1 Requirements for real measurements in the second measurements

In this subchapter, the components and his respective characteristics are presented to realize the real measurements with the balloon.

The carrier frequency in the link between the transmitter and receiver was 2.468 GHz.
First of all, the chosen balloon was the model helical and the transmitter is composed with the omnidirectional antenna, the GPS and the battery. For the GPS, the software used was the GPSD. The components are presented in the next picture:

Figure 3.7: The balloon used

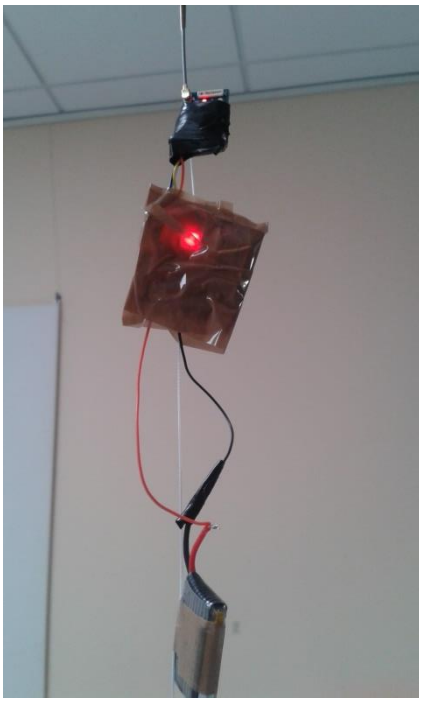


Figure 3.8: From up to down: Omnidirectional antenna, GPS and battery

For the receiver, two models of the antennas are used: Omnidirectional and directional antennas with 2 dB and 7 dB of gain respectively. Every second, 7 samples are received for a total time of 2 minutes in each altitude.

Figure 3.9: The two antennas used

3.2.2 First scenario: in front of the Kasteel van Arenberg Heverlee

In this scenario, the measurements were taken in different altitudes, and for different horizontal distance between the transmitter and the receiver. The balloon always had *line-of-sight* with the ground station. In the following figure it is possible to see how it was the scenario and the point where the measurements were taken:

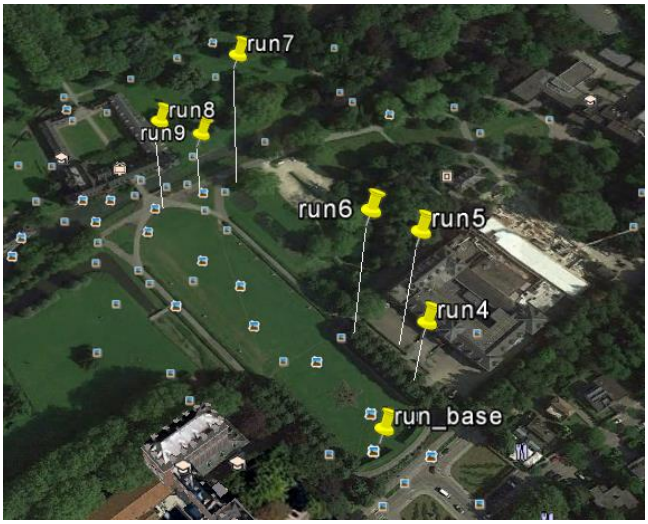


Figure 3.10: First scenario with the balloon in ESAT

First of all, the path loss exponents are calculated from the measurements for each type of antenna. For the following figures are used to verify that, as the distance between the balloon with the ground station increase, the power received decrease.

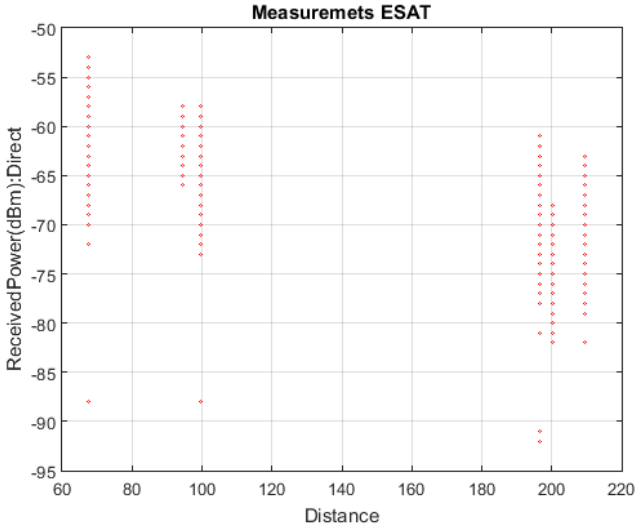


Figure 3.11: Received power with directional antenna in ESAT scenario

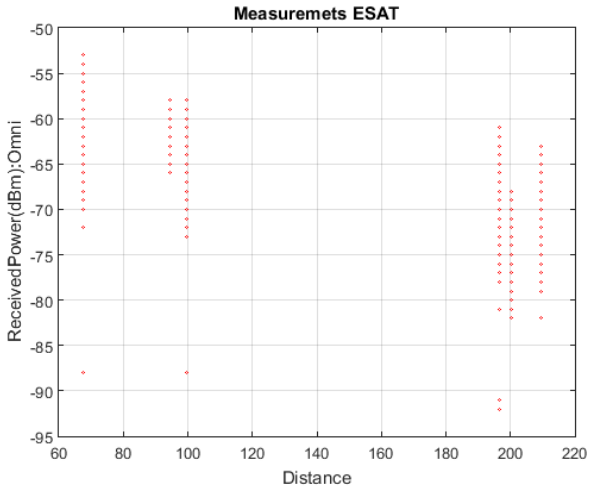


Figure 3.12: Received power with directional antenna in ESAT scenario

Then, the path loss exponent is calculated for each antenna from the path loss equation.
In the table 3.2 the linear model equation shows how we can calculate the path loss exponent $10\alpha = 20.92$ for directional antenna. As it could be expected, the α is very close to the free space attenuation factor (αFS = 2) and, hence, the multiple scatters present in the ESAT scenario do not compromise the path loss fading ratio.

Lineal model equation	α
$f(x) = -21.37 - 20.92log_{10}(d)$	α=2.092

Table 3.2: Path loss exponent for directional antenna

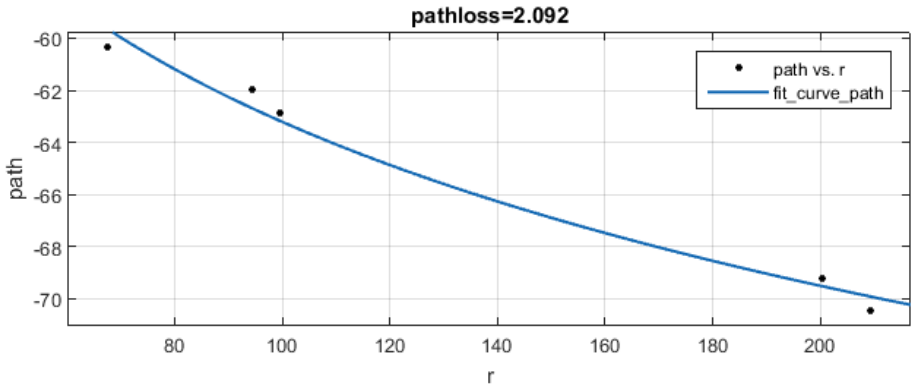


Figure 3.13: Received power as a function of distance for directional antenna

Now, the process is repeated for omnidirectional antenna and the path loss exponent was calculated α=5.016.

Lineal model equation	α
$f(x) = -32.28 - 50.16log_{10}(d)$	α=5.016

Table 3.3: Path loss exponent for omnidirectional antenna

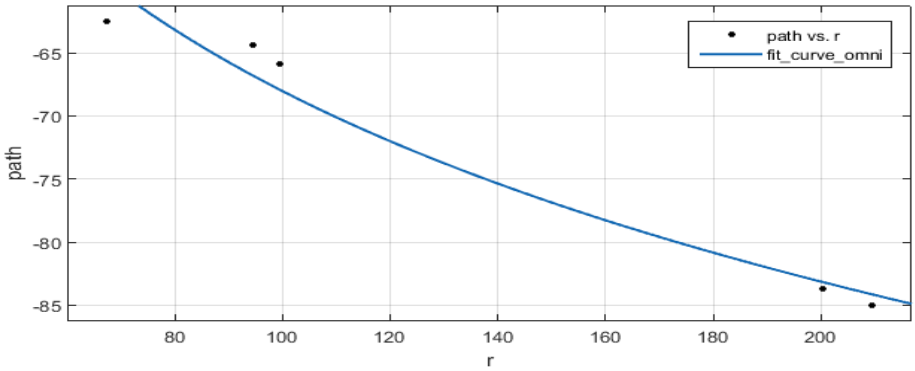


Figure 3.14: Received power as a function of distance for omnidirectional antenna

The difference the value of the path loss exponent between directional antenna and omnidirectional could be for the radiation pattern for each antenna. The omnidirectional could be affected for scattered waves produced by the trees or the buildings behind the antenna and for directional antenna would be less affected [18]. In the next picture, the placing of the antennas is presented in the ESAT scenario:

Figure 3.15: Distribution of the antennas in ESAT scenario

In order to check which Rician K-factor from the measurements data, the received power P_r in Watts has to be converted into voltage V_r in Volts (the envelope amplitude of the received signal) through the equation below because the input of the algorithm is the envelope. A load resistance R_l of 50 Ω has been used in our study.

$$V_r = \sqrt{R_l P_r} \tag{21}$$

In the table 3.4, the estimated Rician K-factor is presented and the moment's method has been used for 2200 samples with series of 50.

Directional antenna

r=58	r=58	r=58	r=194	r=194	r=194
h=34.4	h=74.5	h=80.8	h=78.8	h=49.8	h=32.1
θ=30.67º	θ=52.09º	θ=54.32º	θ=22.1º	θ=14.39º	θ=9.39º
K=16.40	K=61.60	K=35.67	K=23.73	K=49.26	K=11.59

Omnidirectional antenna

r=58	r=58	r=58	r=194	r=194	r=194
h=34.4	h=74.5	h=80.8	h=78.8	h=49.8	h=32.1
θ=30.67º	θ=52.09º	θ=54.32º	θ=22.1º	θ=14.39º	θ=9.39º
K=21.52	K=50.73	K=33.82	K=26.07	K=24.81	K=17.74

Table 3.4: Estimated K's for each type of antenna in ESAT scenario

The main conclusions about the tables are when de angle increase in the scenario, the Rician K-factor also. Moreover, we expected that the values of K in directional antenna were larger than the omnidirectional because the gain of the directive is larger than omnidirectional and the directive should not affect for the scattered waves produced for the buildings, but at some points do not occur this condition.

3.2.3 Second scenario in the street

In this scenario, the measurements were taken in different altitudes, and for different horizontal distance between the transmitter and the receiver. The balloon always had *line-of-sight* with the ground station. In the following figure it is possible to see how it was the scenario and the point where the measurements were taken:

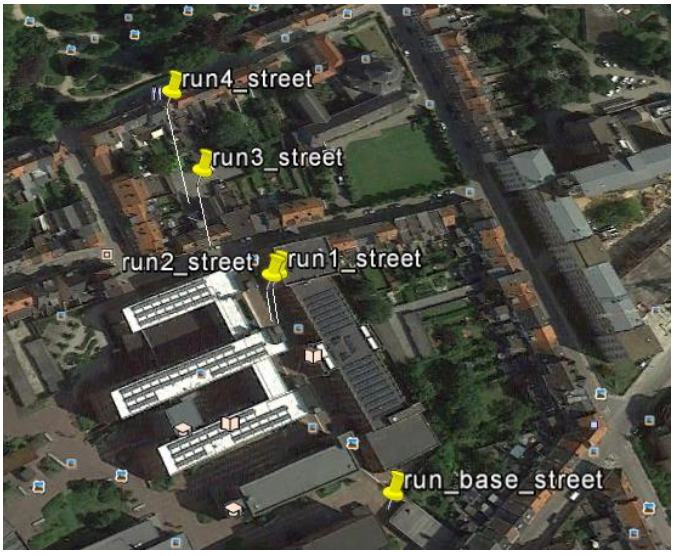


Figure 3.16: Second scenario with the balloon in the street

Now, the same process is repeated as the ESAT scenario to calculate the Rician K-factor we analyse 10.000 samples with series of 100 in each altitude. The values are the following:

Directional antenna

r=85.31	r=85.31	r=139.88	r=139.88
h=30.6	h=26.9	h=44.4	h=60.9
θ=19.73º	θ=17.5º	θ=17.61º	θ=23.52º
K=42	K= 32.98	K= 52.65	K= 62.61

Omnidirectional antenna

r=85.31	r=85.31	r=139.88	r=139.88
h=30.6	h=26.9	h=44.4	h=60.9
θ=19.73º	θ=17.5º	θ=17.61º	θ=23.52º
K=40.12	K= 30.31	K= 42.39	K= 51.12

Table 3.5: Estimated K's for each type of antenna in street scenario

In the table 3.4, the same conclusion can be taken respect to the ESAT scenario and the K's increase when the angle does it. For this case, the values of the directional antenna are larger than de omnidirectional antenna and meet what we expected.

3.2.4 Propose a function about the scenarios with balloon

After analyzing the last two scenarios with the balloon, a function of K respect to angle is presented. To do this, the average K in the different scenarios is calculated and with the fit curve application of the Matlab shows the function for each type of antenna.

Directional antenna

$$f(K) = -183.1\theta^{-0.45} + 77.91$$

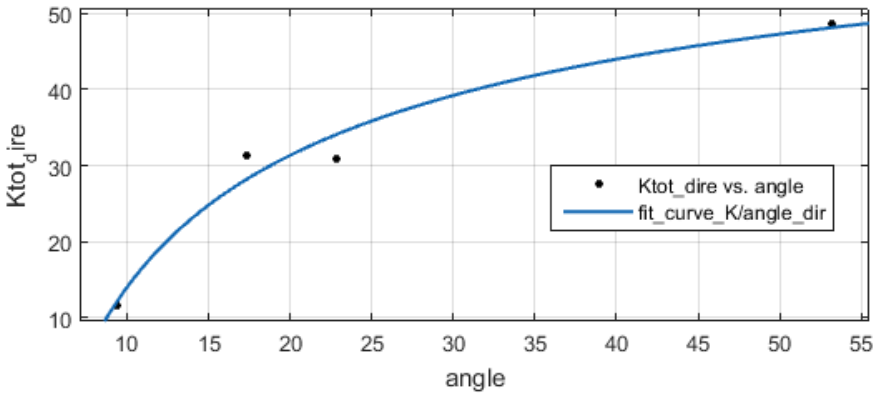


Figure 3.17: Function of K respect to the angle in directional antenna

Omnidirectional antenna

$$f(K) = 11.33\theta^{0.38} - 9.08$$

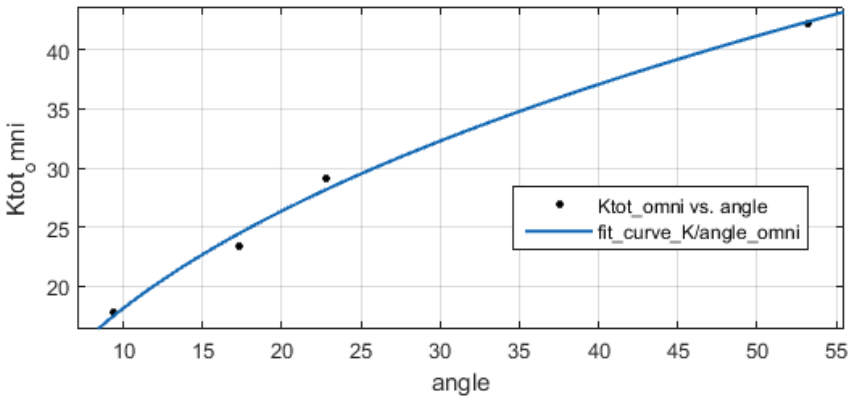


Figure 3.18: Function of K respect to the angle in omnidirectional antenna

Chapter 4: Conclusions

4.1 Conclusions

The reason of this thesis was to study the types of fading and focused in the Rician fading. The aim was to view the behaviour of the key parameter in this type of fading, the Rician K factor with different altitudes and finally propose a function of K respect to the angle.

First of all, the theory propagation about the fading was studied in chapter 2. Besides, in the same chapter the different models like method of moments, maximum likelihood estimator and fitting curve method to estimate the Rician K factor and their respective graphs were presented.
In the last part of this chapter, the comparison between each method was presented and the conclusions have been that the all the methods are biased estimators for finite data samples. The 1st and 2nd moment are the best estimator followed by 2nd and 4th moment, but for large values of K, the best methods are fitting curve and maximum likelihood. Nevertheless, all of these methods are closed.

The second part of the thesis was the empirical results from the measurement in the roof of the ESAT building with the UAV and the second part with the balloon. The two scenarios are explained in the Chapter 3.

The conclusions about the first scenario with the UAV may have been involved the constructive or destructive fading. Therefore, the graphs in the first measurements have seen that for each altitude the fading affect in his received power the independent form, such with altitude of 4 meters is affected for destructive fading and in 6 meters the invers.
To estimate the Rician K-factor, the different methods have been used and the conclusions are that when the altitude is increase the Rician K-factor also. In addition, the values of K's are larger because the distance between the transmitter and receiver was not too long.

The main conclusions in the second part of the empirical measurements with the balloon were that when the angle increases, the Rician K-factor also increase.

Normally for directional antenna the values of the Rician K-factor are larger than the omnidirectional antenna.

4.2 Future work

After the study the empirical measurements and take the conclusions, it would be recommended to analyze the same system in different places with more buildings and in non-line-of-sight conditions and investigate the Rician K-factor.

Therefore, a function of Rician K-factor respect to the angle, or distance or altitude had to be found.

Appendices

Appendix A :

Matlab source code

A1. Theoretical propagation

```
%%%%%%%%%%%%%%%%%%%%%% Theorical propagation %%%%%%%%%%%%%%%%%%
        clear all;
        close all;

%%%%%%%%%%%%%%%%%%%%%% Rayleigh distribution %%%%%%%%%%%%%%%%%

        sigma_ray= [2,0,0; 0,4,0; 0,0,8];

        N=1e6;
        x=randn(N,3)*sigma_ray;      %We operate the matrix
        y=randn(N,3)*sigma_ray;      %(Nx3)*(3x3)=(Nx3)
        r=sqrt(x.^2+y.^2);

        %Histogram of the new variable
        [pr,xout]=hist(r,100);
        subplot(1,2,1)
        plot(xout,pr/N);
        xlim([0,20]);
        xlabel('r');
        ylabel('Normalized histogram');
        legend('sigma=2','sigma=4','sigma=8');

        %%%%%%%%%%%%%%%%%% Rician distribution %%%%%%%%%%%%%%%%%%%

        sigma_rice=1;
        r=[0:0.01:6];
        s=1;
        rice_pdf= (r./sigma_rice^2).*exp(-
    (r.^2+s^2)./(2*sigma_rice^2)).*besseli(0,(r.*s)/(sigma_rice^2
    ));
```

```
figure
plot(r, rice_pdf,'k');
xlabel('r');
ylabel('Rician PDF');

%%%%%%%%%%%%%% Rician distribution respect respect K %%%%%%%%

K=[-40 6 10 20];
nu_rice=1;
nu=0.0145;
nu_g=3;

A_rice0 = (2*r.*10^(K(:,1)/10))/(nu^2);
B_rice0 = (-(10^(K(:,1)/10))/nu^2)*(r.^2+nu^2);
C_rice0 = besseli(0,(2*r.*10^(K(:,1)/10))/(nu));
Rice0= A_rice0.*exp(B_rice0).*C_rice0;

A_rice11 = (2*r.*10^(K(:,2)/10))/(nu_rice^2);
B_rice11 = (-(10^(K(:,2)/10))/nu_rice^2)*(r.^2+nu_rice^2);
C_rice11 = besseli(0,(2*r.*10^(K(:,2)/10))/(nu_rice));
Rice11= A_rice11.*exp(B_rice11).*C_rice11;

A_rice1 = (2*r.*10^(K(:,3)/10))/(nu_rice^2);
B_rice1 = (-(10^(K(:,3)/10))/nu_rice^2)*(r.^2+nu_rice^2);
C_rice1 = besseli(0,(2*r.*10^(K(:,3)/10))/(nu_rice));
Rice1= A_rice1.*exp(B_rice1).*C_rice1;

A_rice2 = (2*r.*10^(K(:,4)/10))/(nu_g^2);
B_rice2 = (-(10^(K(:,4)/10))/nu_g^2)*(r.^2+nu_g^2);
C_rice2 = besseli(0,(2*r.*10^(K(:,4)/10))/(nu_g));
Rice2= A_rice2.*exp(B_rice2).*C_rice2;

figure
plot(r, Rice0,'k',r,Rice11,'r',r, Rice1,'g',r, Rice2,'b');
title('Rician distribution respect K(lineal)');
xlabel('r');
ylabel('PDF');
legend('K=0','K=4','K=10','K=100');
```

A2. Methods to estimate K

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1st and 2nd moments %%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

signal=ones(1e5, 1);
```

```
5.        Kin=[0 4 20 100];
6.        Kfin=[];
7.        y=[];
8.        matr=[];
9.        size_samples=100;
10.       K_count=0;

11.       for K=Kin
12.       chan = ricianchan(1/1000, 100, K);%generate the rician
   channel chan = ricianchan(ts,fd,k)
13.       rcom = filter(chan, signal);
14.       module = abs(rcom);

15.       for m=0:1:((size(module)/size_samples)-1)%Divide the
   module in packages of 100 samples
16.       l=0;
17.       j=m+1;
18.       for i=(m*100+1):1:(m+1)*100
19.       l=l+1;
20.       matr(j,l)=module(i);%Save the packages in the matrix
21.       end

22.       end
23.       K_count=K_count+1;

24.       for j=1:1:(m+1)
25.       V1=matr(j,:);
26.       V1sum=sum(V1);
27.       E1=V1sum/l;
28.       V2=matr(j,:).^2;
29.       V2sum=sum(V2);
30.       E2=V2sum/l;
31.       x=(E1^2)/E2;
32.       if x<=0.78539
33.       Kest=0;
34.       else
35.       fun = @(K) ((pi*exp(-
   K))/(4*(K+1)))*(((K+1)*besseli(0,(K/2))+K*besseli(1,(K/2)))^2
   )-x;
36.       Kest1=solve(fun);
37.       Kest=abs(Kest1);
38.       end
39.       Kfin(j,K_count) = Kest;%Save the values in the matrix
   Kfin
40.       clear Kest1;
41.       clear Kest;
42.       end
43.       end

44.       close all
45.       figure
46.       histogram(Kfin(:,1),'Normalization','pdf');
47.       title('Kin=0')
48.       legend
49.       xlabel('Estimated K');
50.       ylabel('Probability Density Function');
```

```
51.      figure
52.      histogram(Kfin(:,2),'Normalization','pdf');
53.      title('Kin=4')
54.      legend
55.      xlabel('Estimated K');
56.      ylabel('Probability Density Function');
57.      figure
58.      histogram(Kfin(:,3),'Normalization','pdf');
59.      title('Kin=20')
60.      legend
61.      xlabel('Estimated K');
62.      ylabel('Probability Density Function');
63.      figure
64.      histogram(Kfin(:,4),'Normalization','pdf');
65.      title('Kin=100')
66.      legend
67.      xlabel('Estimated K');
68.      ylabel('Probability Density Function');


69.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70.      %%%%%%%%%%%%%%% 2nd and 4th moments %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
71.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72.
73.      clear all;
74.
75.      signal=ones(1e5, 1);
76.      Kin=[0 4 20 100];
77.      Kfin=[];
78.      y=[];
79.      matr=[];
80.      size_samples=100;
81.      K_count=0;
82.      for K=Kin
83.          chan = ricianchan(1/1000, 100, K); %chan =
   ricianchan(ts,fd,k)
84.          rcom = filter(chan, signal);
85.          module = abs(rcom);
86.
87.          for m=0:1:((size(module)/size_samples)-1)%Divide the
   module in packages of 100 samples
88.              l=0;
89.              j=m+1;
90.              for i=(m*100+1):1:(m+1)*100
91.                  l=l+1;
92.                  matr(j,l)=module(i);%Save the packages in
   the matrix
93.              end
94.
95.          end
96.          K_count=K_count+1;
97.          for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
98.           V2=matr(j,:).^2;
99.           V2sum=sum(V2);
100.          E2=V2sum/l;
```

```
101.          V4=matr(j,:).^4;
102.          V4sum=sum(V4);
103.          E4=V4sum/l;
104.          Kest1=(-2*E2^2+E4-E2*sqrt(2*E2^2-E4))/(E2^2-E4);
105.          Kest=abs(Kest1);
106.          Kfin(j,K_count) = Kest; %Save the values in the
   matrix Kfin
107.
108.          end
109.      end
110.      close all
111.      figure
112.      histogram(Kfin(:,1),'Normalization','pdf');
113.      title('Kin=0')
114.      legend
115.      xlabel('Estimated K');
116.      ylabel('Probability Density Function');
117.      figure
118.      histogram(Kfin(:,2),'Normalization','pdf');
119.      title('Kin=4')
120.      legend
121.      xlabel('Estimated K');
122.      ylabel('Probability Density Function');
123.      figure
124.      histogram(Kfin(:,3),'Normalization','pdf');
125.      title('Kin=20')
126.      legend
127.      xlabel('Estimated K');
128.      ylabel('Probability Density Function');
129.      figure
130.      histogram(Kfin(:,4),'Normalization','pdf');
131.      title('Kin=100')
132.      legend
133.      xlabel('Estimated K');
134.      ylabel('Probability Density Function');


135.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
136.      %%%%%%%%%%%%%%%% Maximum Likelihood Estimator %%%%%%%%%%
137.      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
138.
139.      clear all;
140.
141.      signal=ones(1e5, 1);
142.      Kin=[0 4 20 100];
143.      Kfin=[];
144.      matr=[];
145.      size_samples=100;
146.      K_count=0;
147.      for K=Kin
148.          chan = ricianchan(1/1000, 100, K); %chan =
   ricianchan(ts,fd,k)
149.          rcom = filter(chan, signal);
150.          module = abs(rcom);
151.
```

```
152.          for m=0:1:((size(module)/size_samples)-1)%Divide the
   module in packages of 100 samples
153.              l=0;
154.              j=m+1;
155.              for i=(m*100+1):1:(m+1)*100
156.                  l=l+1;
157.                  matr(j,l)=module(i);%Save the packages in
   the matrix
158.              end
159.
160.          end
161.          K_count=K_count+1;
162.          for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
163.              y=(mean(matr(j,:)).^2);
164.              y1=(mean(matr(j,:)));
165.              p=((matr(j,:)-y1).^2);
166.              p1=sum(p);
167.              Kest=(l*y)/(2*p1);
168.              Kfin(j,K_count) = Kest; %Save the values in the
   matrix Kfin
169.
170.          end
171.      End
172.
173.      close all
174.
175.      figure
176.      histogram(Kfin(:,1),'Normalization','pdf');
177.      title('Kin=0')
178.      legend
179.      xlabel('Estimated K');
180.      ylabel('Probability Density Function');
181.      figure
182.      histogram(Kfin(:,2),'Normalization','pdf');
183.      title('Kin=4')
184.      legend
185.      xlabel('Estimated K');
186.      ylabel('Probability Density Function');
187.      figure
188.      histogram(Kfin(:,3),'Normalization','pdf');
189.      title('Kin=20')
190.      legend
191.      xlabel('Estimated K');
192.      ylabel('Probability Density Function');
193.      figure
194.      histogram(Kfin(:,4),'Normalization','pdf');
195.      title('Kin=100')
196.      legend
197.      xlabel('Estimated K');
198.      ylabel('Probability Density Function');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%% Fitting curve %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

signal=ones(1e5, 1);
Kin=[0 4 20 100];
Kfin=[];
matr=[];
size_samples=100;
K_count=0;

for K=Kin
    chan = ricianchan(1/1000, 100, K); %chan =
ricianchan(ts,fd,k)
    rcom = filter(chan, signal);
    module = abs(rcom);

    for m=0:1:((size(module)/size_samples)-1)%Divide the
module in packages of 100 samples
        l=0;
        j=m+1;
        for i=(m*100+1):1:(m+1)*100
            l=l+1;
            matr(j,l)=module(i);%Save the packages in
the matrix
        end

    end
    K_count=K_count+1;
     for j=1:1:(m+1)%Calculate every 100 samples, the
variance and the mean to apply the function
       y=var(matr(j,:));
       x=mean(matr(j,:));
       Kest=(x.^2)/(2*y);

       Kfin(j,K_count) = Kest; %Save the values in the
matrix Kfin

     end
end

close all

figure
histogram(Kfin(:,1),'Normalization','pdf');
title('Kin=0')
legend
xlabel('Estimated K');
ylabel('Probability Density Function');
figure
histogram(Kfin(:,2),'Normalization','pdf');
title('Kin=4')
legend
```

```
249.     xlabel('Estimated K');
250.     ylabel('Probability Density Function');
251.     figure
252.     histogram(Kfin(:,3),'Normalization','pdf');
253.     title('Kin=20')
254.     legend
255.     xlabel('Estimated K');
256.     ylabel('Probability Density Function');
257.     figure
258.     histogram(Kfin(:,4),'Normalization','pdf');
259.     title('Kin=100')
260.     legend
261.     xlabel('Estimated K');
262.     ylabel('Probability Density Function');
```

A3. First measurements in the roof of the ESAT building

```
1.       clear all;
2.       close all;

3.       num_samples = 6e3;
4.       fileID= fopen('0m.bin');
5.       a0= fread(fileID,'int8');
6.       b0=a0(1:2:num_samples)*j + a0(2:2:num_samples);
7.       fclose(fileID);
8.       fileID= fopen('4m.bin');
9.       a1= fread(fileID,'int8');
10.      b1=a1(1:2:num_samples)*j + a1(2:2:num_samples);
11.      fclose(fileID);
12.      fileID= fopen('6m.bin');
13.      a2= fread(fileID,'int8');
14.      b2=a2(1:2:num_samples)*j + a2(2:2:num_samples);
15.      fclose(fileID);
16.      fileID= fopen('8m.bin');
17.      a3= fread(fileID,'int8');
18.      b3=a3(1:2:num_samples)*j + a3(2:2:num_samples);
19.      fclose(fileID);
20.      fileID= fopen('10m.bin');
21.      a4= fread(fileID,'int8');
22.      b4=a4(1:2:num_samples)*j + a4(2:2:num_samples);
23.      fclose(fileID);
24.      %Load the measurements

25.      all_meas=[b0 b1 b2 b3 b4];%real_measurements in
   microVolts
26.      distance = 5; %distance in meters between transmitter
   and receiver
27.      altitude = [0 4 6 8 10];%altitudes of UAV
28.      h=[];
29.      v(:,1)=(abs(all_meas(:,1)*1e-6));%voltage 0m
30.      v(:,2)=(abs(all_meas(:,2)*1e-6));%voltage 4m
31.      v(:,3)=(abs(all_meas(:,3)*1e-6));%voltage 6m
32.      v(:,4)=(abs(all_meas(:,4)*1e-6));%voltage 8m
```

```
33.      v(:,5)=(abs(all_meas(:,5)*1e-6));%voltage 10m

34.      power=((v.^2)/50);%power with Rl=50
35.      powerdBm= 10*log10(power)+30;

36.      h= sqrt((distance^2)+(altitude.^2)); %calculate the
   total distance

37.      %%%%%%% START CALCULATE POWER RECEIVED %%%%%%%%%%%%%%%%%

38.      for i=1:1:length(powerdBm(:,1))
39.      plot(h,powerdBm(i,:),'ro', 'MarkerSize',1.3); hold on

40.      end
41.      ylabel('ReceivedPower(dBm)');
42.      xlabel('Distance(m)');
43.      title('Measuremets (0m/4m/6m/8m/10m)');

44.      grid on
45.      xlim([4 12]);

46.      %%%%%%%%%%%%%%%%%% START ALGORITHM METHOD %%%%%%%%%%%%%
47.      %%%%%%%%%%%%%%%%%% 1st and 2nd moments %%%%%%%%%%%%%%%%
48.
49.      Kfin=[];
50.      y=[];
51.      matr=[];
52.      size_samples=100;
53.      K_count=0;
54.
55.      for m=0:1:((size(vi)/size_samples)-1)%Divide the module
   in packages of 100 samples vi is the voltage for each
   altitude
56.              l=0;
57.              j=m+1;
58.              for i=(m*100+1):1:(m+1)*100
59.                  l=l+1;
60.                  matr(j,l)=vi(i);%Save the packages in the
   matrix
61.              end
62.
63.      end
64.          K_count=K_count+1;
65.      for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
66.           V1=matr(j,:);
67.           V1sum=sum(V1);
68.           E1=V1sum/l;
69.           V2=matr(j,:).^2;
70.           V2sum=sum(V2);
71.           E2=V2sum/l;
72.           x=(E1^2)/E2;
73.           if x<=0.78539
74.               Kest=0;
75.           else
```

```
76.                 fun = @(K) ((pi*exp(-
   K))/(4*(K+1)))*(((K+1)*besseli(0,(K/2))+K*besseli(1,(K/2)))^2
   )-x;
77.                 Kest1=solve(fun);
78.                 Kest=abs(Kest1);
79.             end
80.             Kfin(j,K_count) = Kest;%Save the values in the
   matrix Kfin
81.             clear Kest1;
82.             clear Kest;
83.
84.         end


85.         %%%%%%%%%%%%%%%% START ALGORITHM METHOD %%%%%%%%%%%%%%%
86.         %%%%%%%%%%%%%%%% 2nd and 4th moments %%%%%%%%%%%%%%%%%%%%
87.
88.         Kfin=[];
89.         y=[];
90.         matr=[];
91.         size_samples=100;
92.         K_count=0;

93.         for m=0:1:((size(vi)/size_samples)-1)%Divide the module
   in packages of 100 samples vi is the voltage for each
   altitude
94.         l=0;
95.         j=m+1;
96.         for i=(m*100+1):1:(m+1)*100
97.         l=l+1;
98.         matr(j,l)=vi(i);%Save the packages in the matrix
99.         end

100.        end
101.        K_count=K_count+1;
102.        for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
103.        V2=matr(j,:).^2;
104.             V2sum=sum(V2);
105.             E2=V2sum/l;
106.             V4=matr(j,:).^4;
107.             V4sum=sum(V4);
108.             E4=V4sum/l;
109.             Kest1=(-2*E2^2+E4-E2*sqrt(2*E2^2-E4))/(E2^2-E4);
110.             Kest=abs(Kest1);
111.        end
112.
113.        %%%%%%%%%%%%%%%% START ALGORITHM METHOD %%%%%%%%%%%%%%%
114.        %%%%%%%%%% Maximum Likelihood Estimator %%%%%%%%%%%%%%%%
115.
116.        Kfin=[];
117.        y=[];
118.        matr=[];
119.        size_samples=100;
120.        K_count=0;
```

```
121.     for m=0:1:((size(vi)/size_samples)-1)%Divide the module
   in packages of 100 samples vi is the voltage for each
   altitude
122.     l=0;
123.     j=m+1;
124.     for i=(m*100+1):1:(m+1)*100
125.     l=l+1;
126.     matr(j,l)=vi(i);%Save the packages in the matrix
127.     end

128.     end
129.     K_count=K_count+1;
130.     for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
131.           y=(mean(matr(j,:)).^2);
132.           y1=(mean(matr(j,:)));
133.           p=((matr(j,:)-y1).^2);
134.           p1=sum(p);
135.           Kest=(l*y)/(2*p1);
136.           Kfin(j,K_count) = Kest;
137.     end
138.
139.     %%%%%%%%%%%%%%%% START ALGORITHM METHOD %%%%%%%%%%%%%%%
140.     %%%%%%%%%%%%%%%% Fitting curve %%%%%%%%%%%%%%%%%%%%%%%%
141.
142.     Kfin=[];
143.     y=[];
144.     matr=[];
145.     size_samples=100;
146.     K_count=0;

147.     for m=0:1:((size(vi)/size_samples)-1)%Divide the module
   in packages of 100 samples vi is the voltage for each
   altitude
148.     l=0;
149.     j=m+1;
150.     for i=(m*100+1):1:(m+1)*100
151.     l=l+1;
152.     matr(j,l)=vi(i);%Save the packages in the matrix
153.     end

154.     end
155.     K_count=K_count+1;
156.     for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
157.          y=var(matr(j,:));
158.          x=mean(matr(j,:));
159.          Kest=(x.^2)/(2*y);
160.          Kfin(j,K_count) = Kest;
161.     End
```

A4. Second measurements in the ESAT garden

```
1.    clear all;
2.    clc;

3.    num_samples = 2200;

4.    fileID = fopen('run4_direct.wifi');
5.    a4_direct = fscanf(fileID,'%i');
6.    b4_direct=a4_direct(1:1:num_samples);
7.    fileID = fopen('run4_omni.wifi');
8.    a4_omni = fscanf(fileID,'%i');
9.    b4_omni=a4_omni(1:1:num_samples);

10.   fileID = fopen('run5_direct.wifi');
11.   a5_direct = fscanf(fileID,'%i');
12.   b5_direct=a5_direct(1:1:num_samples);
13.   fileID = fopen('run5_omni.wifi');
14.   a5_omni = fscanf(fileID,'%i');
15.   b5_omni=a5_omni(1:1:num_samples);

16.   fileID = fopen('run6_direct.wifi');
17.   a6_direct = fscanf(fileID,'%i');
18.   b6_direct=a6_direct(1:1:num_samples);
19.   fileID = fopen('run6_omni.wifi');
20.   a6_omni = fscanf(fileID,'%i');
21.   b6_omni=a6_omni(1:1:num_samples);

22.   fileID = fopen('run7_direct.wifi');
23.   a7_direct = fscanf(fileID,'%i');
24.   b7_direct=a7_direct(1:1:num_samples);
25.   fileID = fopen('run7_omni.wifi');
26.   a7_omni = fscanf(fileID,'%i');
27.   b7_omni=a7_omni(1:1:num_samples);

28.   fileID = fopen('run8_direct.wifi');
29.   a8_direct = fscanf(fileID,'%i');
30.   b8_direct=a8_direct(1:1:num_samples);
31.   fileID = fopen('run8_omni.wifi');
32.   a8_omni = fscanf(fileID,'%i');
33.   b8_omni=a8_omni(1:1:num_samples);

34.   fileID = fopen('run9_direct.wifi');
35.   a9_direct = fscanf(fileID,'%i');
36.   b9_direct=a9_direct(1:1:num_samples);
37.   fileID = fopen('run9_omni.wifi');
38.   a9_omni = fscanf(fileID,'%i');
39.   b9_omni=a9_omni(1:1:num_samples);

40.   p_dbm_direct = [b4_direct,b5_direct, b6_direct,
   b7_direct, b8_direct, b9_direct];
41.   p_dbm_omni = [b4_omni,b5_omni, b6_omni, b7_omni,
   b8_omni, b9_omni];
```

```
42.   r = [58, 194];
43.   altitude1 = [34.4, 74.5, 80.8];
44.   altitude2 = [78.8, 49.8, 32.1];
45.   altitude=[altitude1 altitude2];

46.   for i=1:3
47.   distance1(i)= sqrt(r(1)^2+altitude1(i)^2);
48.   angle_rad1(i) = atan(altitude1(i)/r(1));
49.   angle_deg1(i) = angle_rad1(i)*(180/pi);
50.   end
51.   for i=1:3
52.   distance2(i)= sqrt(r(2)^2+altitude2(i)^2);
53.   angle_rad2(i) = atan(altitude2(i)/r(2));
54.   angle_deg2(i) = angle_rad2(i)*(180/pi);
55.   end
56.
57.   figure
58.   for i=1:1:length(p_dbm_direct(:,1))
59.       plot(distance,p_dbm_direct(i,:),'ro',
   'MarkerSize',1.3); hold on
60.
61.   end
62.   ylabel('ReceivedPower(dBm):Direct');
63.   xlabel('Distance');
64.   title('Measuremets ESAT');
65.   grid on
66.   xlim([60 220]);
67.
68.   figure
69.   for i=1:1:length(p_dbm_omni(:,1))
70.       plot(distance,p_dbm_direct(i,:),'ro',
   'MarkerSize',1.3); hold on
71.
72.   end
73.   ylabel('ReceivedPower(dBm):Omni');
74.   xlabel('Distance');
75.   title('Measuremets ESAT');
76.   grid on
77.   xlim([60 220]);
78.
79.   distance=[distance1,distance2];
80.   angle_deg=[angle_deg1,angle_deg2];
81.   for i=1:6
82.   p_mw_direct(:,i)=10.^(p_dbm_direct(:,i)/10);
83.   p_mw_omni(:,i)=10.^(p_dbm_omni(:,i)/10);
84.   p_w_direct(:,i)= p_mw_direct(:,i)/1000;
85.   p_w_omni(:,i)= p_mw_omni(:,i)/1000;
86.   v_direct(:,i) = sqrt(2*50*p_w_direct(:,i));
87.   v_omni(:,i) = sqrt(2*50*p_w_omni(:,i));
88.   end
89.
90.   module = v_omni(:,1);
91.   Kfin=[];
92.   y=[];
93.   matr=[];
94.   size_samples=100;
```

```
95.   K_count=0;
96.
97.   for m=0:1:((size(module)/size_samples)-1)%Divide the
   module in packages of 100 samples
98.           l=0;
99.           j=m+1;
100.          for i=(m*100+1):1:(m+1)*100
101.              l=l+1;
102.              matr(j,l)=module(i);%Save the packages in
   the matrix
103.          end
104.
105.  end
106.      K_count=K_count+1;
107.  for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
108.       V2=matr(j,:).^2;
109.       V2sum=sum(V2);
110.       E2=V2sum/l;
111.       V4=matr(j,:).^4;
112.       V4sum=sum(V4);
113.       E4=V4sum/l;
114.       Kest1=(-2*E2^2+E4-E2*sqrt(2*E2^2-E4))/(E2^2-E4);
115.       Kest=abs(Kest1);
116.       Kfin(j,K_count) = Kest; %Save the values in the
   matrix Kfin
117.
118.  End
119.
120.  Kdirec=[16.40 61.60 35.67 23.73 49.26 11.59];
121.  K_direc1=[20.05 49.13 11.59];
122.  angle_direc=[26.38 53 9.39];
123.
124.  K_omni=[21.52 50.73 33.82 26.07 24.81 17.74];
125.  K_omni1=[24.13 42.27 17.74];
126.  angle_omni=[22.39 53 9.39];
127.
128.  K1_dir=11.59;
129.  K2_dir=[49.26 26.06 16.89 33.1];
130.  K2_dir=mean(K2_dir);
131.  K3_dir=[30.89];
132.  K3_dir=mean(K3_dir);
133.  K5_dir=[61.6 35.67];
134.  K5_dir=mean(K5_dir);
135.  Ktot_dire=[K1_dir K2_dir K3_dir K5_dir];
136.  angle=[9.39 17.3 22.81 53.2];
137.  Ktot_omni=[17.74 23.43 29.06 42.27];
138.  figure
139.  plot(angle, Ktot_omni, 'bo','MarkerSize',8);
140.  legend
141.  xlabel('Angle');
142.  ylabel('K');
143.  grid on
144.  xlim([0 60]);
145.  figure
146.  plot(angle, Ktot_dire, 'bo','MarkerSize',8);
```

```
147.  legend
148.  xlabel('Angle');
149.  ylabel('K');
150.  grid on
151.  xlim([0 60]);
```

A5. Second measurements in the street

```
1.        clc;
2.        clear all;
3.        num_samples=10e3;
4.
5.        fileID = fopen('run_1direct.wifi');
6.        a1_direct = fscanf(fileID,'%i');
7.        b1_direct=a1_direct(1:1:num_samples);
8.        fileID = fopen('run_1omni.wifi');
9.        a1_omni = fscanf(fileID,'%i');
10.            b1_omni= a1_omni(1:1:num_samples);
11.
12.       fileID = fopen('run_2direct.wifi');
13.       a2_direct = fscanf(fileID,'%i');
14.       b2_direct=a2_direct(1:1:num_samples);
15.       fileID = fopen('run_2omni.wifi');
16.       a2_omni = fscanf(fileID,'%i');
17.       b2_omni=a2_omni(1:1:num_samples);
18.
19.       fileID = fopen('run_3direct.wifi');
20.       a3_direct = fscanf(fileID,'%i');
21.       b3_direct=a3_direct(1:1:num_samples);
22.       fileID = fopen('run_3omni.wifi');
23.       a3_omni = fscanf(fileID,'%i');
24.       b3_omni=a3_omni(1:1:num_samples);
25.
26.       fileID = fopen('run_4direct.wifi');
27.       a4_direct = fscanf(fileID,'%i');
28.       b4_direct=a4_direct(1:1:num_samples);
29.       fileID = fopen('run_4omni.wifi');
30.       a4_omni = fscanf(fileID,'%i');
31.       b4_omni=a4_omni(1:1:num_samples);
32.
33.       p_dbm_direct = [b1_direct, b2_direct, b3_direct,
    b4_direct];
34.       p_dbm_omni = [b1_omni, b2_omni, b3_omni, b4_omni];
35.
36.       r = [85.31, 139.88];
37.       altitude1 = [30.6, 26.9];
38.       altitude2 = [44.4, 60.9];
39.       altitude=[altitude1 altitude2];
40.
41.       for i=1:2
42.       distance1(i)= sqrt(r(1)^2+altitude1(i)^2);
43.       angle_rad1(i) = atan(altitude1(i)/r(1));
```

```
44.       angle_deg1(i) = angle_rad1(i)*(180/pi);
45.       end
46.       for i=1:2
47.       distance2(i)= sqrt(r(2)^2+altitude2(i)^2);
48.       angle_rad2(i) = atan(altitude2(i)/r(2));
49.       angle_deg2(i) = angle_rad2(i)*(180/pi);
50.       end
51.
52.       distance=[distance1,distance2];
53.       angle_deg=[angle_deg1,angle_deg2];
54.       figure
55.       for i=1:1:length(p_dbm_direct(:,1))
56.           plot(distance,p_dbm_direct(i,:),'ro',
    'MarkerSize',1.3); hold on
57.
58.       End
59.       ylabel('ReceivedPower(dBm):Direct');
60.       xlabel('Distance');
61.       title('Measuremets street');
62.       grid on
63.       xlim([80 160]);
64.
65.       figure
66.       for i=1:1:length(p_dbm_omni(:,1))
67.           plot(distance,p_dbm_omni(i,:),'ro',
    'MarkerSize',1.3); hold on
68.
69.       end
70.
71.       ylabel('ReceivedPower(dBm):Omni');
72.       xlabel('Distance');
73.       title('Measuremets street');
74.       grid on
75.       xlim([80 160]);
76.       for i=1:4
77.       p_mw_direct(:,i)=10.^(p_dbm_direct(:,i)/10);
78.       p_mw_omni(:,i)=10.^(p_dbm_omni(:,i)/10);
79.       p_w_direct(:,i)= p_mw_direct(:,i)/1000;
80.       p_w_omni(:,i)= p_mw_omni(:,i)/1000;
81.       v_direct(:,i) = sqrt(2*50*p_w_direct(:,i));
82.       v_omni(:,i) = sqrt(2*50*p_w_omni(:,i));
83.       end
84.       module =v_omni(:,1);
85.       Kfin=[];
86.       y=[];
87.       matr=[];
88.       size_samples=50;
89.       K_count=0;
90.
91.       for m=0:1:((size(module)/size_samples)-1)%Divide the
    module in packages of 100 samples
92.                 l=0;
93.                 j=m+1;
94.                 for i=(m*50+1):1:(m+1)*50
95.                     l=l+1;
```

```
                matr(j,l)=module(i);%Save the packages in
   the matrix
            end

        end
            K_count=K_count+1;
        for j=1:1:(m+1)%Calculate every 100 samples, the
   variance and the mean to apply the function
          V2=matr(j,:).^2;
          V2sum=sum(V2);
          E2=V2sum/l;
          V4=matr(j,:).^4;
          V4sum=sum(V4);
          E4=V4sum/l;
          Kest1=(-2*E2^2+E4-E2*sqrt(2*E2^2-E4))/(E2^2-E4);
          Kest=abs(Kest1);
          Kfin(j,K_count) = Kest; %Save the values in the
   matrix Kfin

        end
```

Bibliography

[1] "Unmanned aerial vehicles (UAV)" https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

[2] Fabio Dovis, Roberto Fantini, Marina Mondin, and Patrizia Savi, "*Small-Scale Fading for High-Altitude Platform (HAP) Propagation Channels*"

[3] Sateaa H.alnajjar *"Low-Altitude Platform to enhance communications reliability in Disaster environments*"

[4] Andrea Goldsmith " *Wireless communications*" Standford University

[5] David Tse and Pramod Viswanath "Fundamentals Wireless Communication chapter2"

[6] Fabio Belloni "*Fading Models*" S-88 Signal Processing Laboratory, HUT

[7] M.Garcia, LCSF (Wirelles communications Laboratory) Theory slides, Chapter 1- Caracterización de la propagación en el medio radio. UPC.

[8] Sanjay Kumar, "*Wireless Communication the fundamental and advanced concepts*"

[9] "Rayleigh fading" http://www.radio-electronics.com/info/propagation/multipath/rayleigh-fading-tutorial-basics.php

[10] F. Fontán and P.Espineira, "*Modelling the wireless propagation channel"*

[11] Cihan Tepedelenlio˘glu, Member, IEEE, Ali Abdi, Member, IEEE, and Georgios B. Giannakis, Fellow, IEEE , "*The Ricean K Factor: Estimation and performance analysis"*

[12] Geoffrey G. Messier, Member, IEEE, and Jennifer A. Hartwell , "*An Empirical Model for Nonstationary Ricean Fading"*

[13] Sathyaveer Prasad1,2, Samer Medawar2, Peter Händel1,2, Claes Beckman 1, "*Estimation of the Rician K-factor in reverberation chambers for improved repeatability in terminal antenna measurements"*

[14] Jan Sijbers,* Arnold J. den Dekker, Paul Scheunders, and Dirk Van Dyck, "*Maximum-Likelihood Estimation of Rician Distribution Parameters"*

[15] "Curve fitting method" https://terpconnect.umd.edu/toh/spectrum/CurveFitting.html

[16] Marvin K.Simon and Mohamed-Slim Alouini, *"Digital Communications over Fading Channels"*

[17] Catherine Forbes, Merran Evans, Nicholas Hastings and Brian Peacock, "Statical Distributions"

[18] "Path loss exponent in different types of environment" https://www.csie.ntu.edu.tw/~hsinmu/courses/_media/wn_11fall/path_loss_and_shadowing.pdf